TIMING MULTIMODAL TURN-TAKING IN HUMAN-ROBOT COOPERATIVE ACTIVITY

A Thesis
Presented to
The Academic Faculty

by

Crystal Chao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
May 2015

TIMING MULTIMODAL TURN-TAKING IN HUMAN-ROBOT COOPERATIVE ACTIVITY

Approved by:

Professor Andrea L. Thomaz, Advisor
School of Interactive Computing
Georgia Institute of Technology

Professor Ronald C. Arkin
School of Interactive Computing
Georgia Institute of Technology

Professor Henrik I. Christensen
School of Interactive Computing
Georgia Institute of Technology

Professor Karen M. Feigh
School of Aerospace Engineering
Georgia Institute of Technology

Professor Candace L. Sidner
Department of Computer Science
Worcester Polytechnic Institute

Date Approved: 6 April 2015

ACKNOWLEDGEMENTS

First, I must thank my amazing advisor, Andrea Thomaz, who has been a beacon of encouragement and wisdom in this quest. I continue to be impressed by her ability to channel my vague and somewhat incoherent ramblings into productive projects. My final experiment was a two-year-long leap of faith that could not be accomplished without her unwavering support.

My committee members have provided invaluable feedback, with diligence and detail far beyond the call of duty: Ron Arkin, Henrik Christensen, Karen Feigh, and Candy Sidner. I am honored to be held to their high standard. A special thanks to Candy Sidner and Chuck Rich for offering continued guidance and inspiration on this topic throughout the years.

My collaborators were essential to conducting this research. Jinhan Lee and Momotaz Begum contributed to development for the Simon Says experiment. Matthew Gombolay and Julie Shah collaborated on developing scheduling techniques for multimodal behavior. Kalesha Bullard and Tesca Fitzgerald helped to recruit participants for the last experiment. Jinhan Lee, Karl Jiang, and Kalesha Bullard assisted with video coding. I also would like to mention Mikhail Jacob and Brian Magerko for conversations about status in acting that partly inspired the floor regulation model in Chapter 6, and Akansel Cosgun and Justin Smith for practical conversations about TPNs that informed the writing of Chapter 10.

The Office of Naval Research supported this work through ONR Young Investigator Award N00014-08-1-0842 and grant N00014-12-1-0484. I thank the program officers Paul Bello and Tom McKenna for their continued commitment to basic research in human-robot interaction and cognitive science.

All current and past members of the Socially Intelligent Machines Lab must be credited for so many stimulating discussions about the nature of intelligence, learning, and interaction. Outside of this thesis, I am grateful to have had the opportunity to develop many domains of interaction for Simon the Robot in collaboration with Michael Gielniak, Maya Cakmak, Jinhan Lee, Nick DePalma, Chien-Ming Huang, Baris Akgun, and Jae Wook Yoo.

The ladies of Robowomen have been wonderful at creating an environment of camaraderie in this program. Hae Won Park, Maya Cakmak, and Tiffany Chen have especially acted as great friends and confidants.

In no particular order: thanks also to Eric Schumacher, whose cognitive psychology course shaped my views on cognition and interaction. To Gil Weinberg and Mason Bretan for enabling me to perform with Shimon the marimba robot during my time at Georgia Tech, a treasured HRI experience. To Mike Stilman, who taught all of us that no obstacle is immovable — you will be missed.

My family has been incredibly supportive through this period. My father has been my role model, my sister has been my best friend, and my lovable tiger mother sacrificed everything to get me to where I am today.

The greatest thanks must go to my husband Alex Trevor, who has also been my partner in this strange doctoral journey. I owe you for luring me to this wild world of robotics, treating me to the most enlightening dinner conversations on earth, and lifting my spirits in the darkest hours. Thank you for your indefatigable patience, and your unconditional love and commitment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Turn-taking is a fundamental process that governs social interaction. When humans interact, they naturally take initiative and relinquish control to each other using verbal and nonverbal behavior in a coordinated manner. In contrast, existing approaches for controlling a robot's social behavior do not explicitly model turn-taking, resulting in interaction breakdowns that confuse or frustrate the human and detract from the dyad's cooperative goals. They also lack generality, relying on scripted behavior control that must be designed for each new domain.

This thesis seeks to enable robots to cooperate fluently with humans by automatically controlling the timing of multimodal turn-taking. Based on our empirical studies of interaction phenomena, we develop a computational turn-taking model that accounts for multimodal information flow and resource usage in interaction. This model is implemented within a novel behavior generation architecture called CADENCE, the Control Architecture for the Dynamics of Embodied Natural Coordination and Engagement, that controls a robot's speech, gesture, gaze, and manipulation. CADENCE controls turn-taking using a timed Petri net (TPN) representation that integrates resource exchange, interruptible modality execution, and modeling of the human user. We demonstrate progressive developments of CADENCE through multiple domains of autonomous interaction encompassing situated dialogue and collaborative manipulation. We also iteratively evaluate improvements in the system using quantitative metrics of task success, fluency, and balance of control.

CHAPTER I

INTRODUCTION

1.1 Motivation

In lofty visions offered by science and science fiction alike, robots are not just busy behind the scenes but also placed in highly visible roles within human environments. Each advance in the field brings us closer towards a reality of robots collaborating with humans in manufacturing, assisting nurses in hospitals, or serving us in the comfort of our homes. Unlike the computers of old, into whose abstractions we delved using GUIs and mice and keyboards, these robots will be immersed in *our* world. They will see our sights, share our space, and touch our things. What protocol will we use to communicate with these machines, that has sufficient power to capture the richness of the physical world and our interactions with it?

In fact, humans already have a framework for exactly this problem. It's called social interaction, an amalgam of natural language, nonverbal behavior, and societal customs that we use to accomplish information transfer and joint action. Humans use social interaction to learn, collaborate, inform, and delegate. The dynamics of social interaction are governed by *turn-taking*, a process that determines when a participant within an interaction is allowed to take action (a turn). Given that humans have a lifetime of experience with social interaction, it seems intuitive to provide robots with the same capabilities if they are required to interface with humans. Ideally, human-robot interactions would achieve the same level of *fluency* as human-human ones do, in which both participants seamlessly and efficiently adapt to each other's timing and decisions when acting to achieve their goals.

Unfortunately, current approaches to robot control result in interactions that do

not live up to this ideal. Robots exhibit rigid, stop-and-go turn-taking that contrasts with the adaptive dance that humans master at a young age. Disfluency can manifest in a variety of ways, including stretches of dead time, unintended overlaps, lack of responsiveness, and confusion from the human about when to act. While users often cannot identify the specific symptoms, they are quick to judge the holistic interaction as "awkward." As we will show later, these breakdowns subsequently detract from accomplishing interaction goals.

One reason for this continued awkwardness is that turn-taking is often relegated to the status of emergent behavior in human-robot interaction (HRI) systems, rather than treated as an interaction process to be explicitly controlled. If a robot does not have the capacity to adapt to the human's style or take the initiative to repair breakdowns, then the onus is on the human to adapt to the robot's incidental turn-taking dynamics that occur as a side effect of the robot's other behaviors. To overcome these problems, it is essential to understand the critical phenomena underlying human turn-taking and then develop explicit control for turn-taking that plays an integral part in the robot's social decision-making process.

However, it is also insufficient simply to develop isolated controllers for these individual phenomena; we must also integrate them. A theme of this work is that appropriate turn-taking is the sum of many cognitive and behavioral factors that must all go right in the moment. Without integration, we are still left with our lowest common denominator of behavior. The approach of this thesis is to build autonomous controllers for turn-taking that are well-defined within the context of a larger architecture for social HRI. This novel architecture is CADENCE, the Control Architecture for the Dynamics of Embodied Natural Coordination and Engagement.

In the remainder of this chapter, we describe the key challenges in modeling turn-taking in Section 1.2, scope the interaction context in Section 1.3, then state the contributions of this thesis in Section 1.4.

1.2 Factors of the turn-taking problem

1.2.1 Reciprocity

The need for turn-taking behavior is a consequence of *reciprocal interaction* — the implicit notion that, in order to maintain cooperation, each participant in the interaction owes something to the other. Reciprocity contrasts with reactive egocentrism as a control paradigm for interactive robots. A robot that performs all of its actions as a reactive response to a human command may lead to the illusion of turn-taking, but this illusion disperses as soon as one encounters any of the overlapping or uncertainty that occurs in ordinary human communication. A human who perceives her partner to be communicating but doesn't understand the content might ask, "What?" or "I'm sorry?" rather than act like her partner never attempted to communicate. Two humans who start speaking at the same time might stop and figure out who should take a turn, rather than each continuing her own turn. Recovery from breakdowns and miscommunication is a fundamental skill in cooperation, and a robot without reciprocal skills makes no contribution towards such recovery, leaving it entirely to the human to guess what is wrong.

That is, humans have expectations about when to expect actions or information, and they resolve their misunderstandings when their expectations are not met. Generally, human interactions are not behavioral pageants stripped of intention, but concerted attempts to reach common ground. In addition, humans know that their interaction partners hold them to these same expectations. To engage in reciprocal behavior, a robot needs to model and predict the needs of its partner. It also requires an ability to take initiative in the absence of explicit human direction — a sense of obligation as to when to take a turn.

In our work, this duality is addressed through modeling the seizing and yielding of shared resources. Turn-taking commonly refers specifically to the fluent exchange of the speaking floor. We believe that the embodied nature of turn-taking in HRI

necessitates a broader perspective, in which turn-taking is a phenomenon that arises in the presence of any bottlenecking resources that are exchanged by interaction participants. These include the speaking floor in addition to shared resources in other embodied modalities, such as control over shared physical space or objects. The robot needs to be aware of and manage these shared resources while exhibiting reciprocal behavior.

1.2.2 Multimodality

Turn-taking in embodied interaction is also a *multimodal* process. Gaze, gesture, and speech are used for communicative purposes, and these are layered on top of instrumental actions such as manipulation. These modalities are applicable or preferred in different situations, and sometimes must be synchronized, as when referring to something in the environment. The robot must both be able to control its behavior in these modalities as well as perceive multimodal cues from the human that are noisy and unreliable.

In CADENCE, actions across modalities can be executed concurrently or coordinated as needed. Multimodal perception of the human is performed within a user model. The robot's multimodal actions depend on the availability of resources that are shared with the user.

1.2.3 Timing

The turn-taking problem also has sensitive *timing* requirements. All computations in an interaction must be real-time, as latency differences on the order of hundreds of milliseconds are noticeable to humans. Next events are produced largely as a function of time since a previous event occurred, rather than only based on state. Embodied actions are also temporally extended and can differ in duration based on physical limits, as well as be delayed based on differences in cognitive processing. Our work aims to account for these factors while retaining the responsiveness and immediacy

of action execution found in reactive systems [17, 3].

Timing can also affect interactions at a holistic level. One clear example is any domain where efficient execution is desirable. In these cases, slow response times or idle times are problematic, and concurrency across agents or modalities is beneficial. In addition, people gradually synchronize the timing of communicative behaviors to interaction partners over time [19]. Imbalances in the amount of time taken by each interaction partner lead to perceptions of power; for example, speaking quickly and over shorter durations than one's partner can convey lower dominance over the conversational floor. Altering the balance of control can affect the decisions that people make in a social situation, which in turn can affect the outcome of a task.

Many existing interaction control paradigms focus on sequential decision-making, which seeks to perform actions in the correct order. For example, robots can be programmed to make plans and select the correct response in a dialogue. In addition to this, we want to enable robots to manage how they time their actions. Often, the default timing for an action is "however long it takes" – an action starts whenever the computation that produces it finishes, and the action always runs to completion. For reasons listed previously, though, this may not always be the appropriate decision. Managing timing means having an understanding of the impact that timing changes can have on interactions, as well as having a control system that enables the manipulation of timing changes with enough flexibility. Our work enables action concurrency, synchronization, interruption, incremental execution, and parametrized temporal constraints to achieve more natural turn-taking timing.

1.3 Interaction context

Here we briefly describe the entering assumptions of this research. At a high-level, we view an interaction as being in one of three stages: engagement, regulation, and

disengagement. This thesis focuses on the middle stage of regulation, of which turn-taking is a major part. The problem of continuously monitoring when participants have engaged and disengaged from the turn-taking interaction is omitted.

We also focus solely on *dyadic* interactions, in which there is a single human and a single robot. While many of our results are applicable to multiparty interactions, we do not explicitly address any such extensions. The dyad is also presumed to be participating in a shared cooperative activity as defined by Bratman [15]. This formulation stands in contrast to competitive or adversarial interactions, in which participants are not achieving shared or compatible goals.

The multimodal aspect of the work focuses on the robot modalities of speech, gaze, gesture, manipulation while also monitoring those of the human. All of this research is conducted using the upper-torso humanoid robot Simon, designed and built by Meka Robotics. Simon has two 7-DOF arms with 4-DOF hands, which are used for gesturing in addition to picking, placing, and pointing at objects. The arms and hands have compliant series-elastic actuators for safe operation near a human. Simon also has a socially expressive head and neck for gaze and head gestures. The head includes two eyes and two ears, the latter of which contain colored LED arrays. Speakers near Simon's base are used for communicating through text-to-speech. Simon does not have a mobile base, so interactions are conducted face-to-face in a single laboratory environment. The robot hardware is controlled using C6 (Creatures 6), an earlier version of which is described in [10].

1.4 Contributions and overview

This work in this thesis is presented in chronological order and is intended to be read as such. The bulk of the thesis describes the iterative development and evaluation of CADENCE, and each chapter is motivated largely by shortcomings of previous iterations.

1.4.1 Thesis statement

> The ability to control the timing of multimodal turn-taking enables robots to cooperate more fluently and effectively with humans.

This statement is supported through the following contributions:

1. The development of a computational model for controlling the timing of turn-taking.
2. The development of an architecture, CADENCE, for generating multimodal reciprocal behavior for interaction.
3. The demonstration of autonomous interactions with humans in multiple cooperative domains using CADENCE.
4. The evaluation of the benefits of the system with respect to fluency, balance of control, and task success.

Next, an overview of the document is given in terms of supporting each of these contributions.

1.4.2 Contribution 1: Turn-taking model

Chapter 3 introduces the concepts of *minimum necessary information* and information flow, which affect the timing of responses and interruptions. These ideas are further developed in Chapter 8, which elaborates the relationship between semantics and turn-taking in embodied dialogue.

Our turn-taking model also emphasizes a balance between yielding resources, presented in Chapter 5, and seizing resources, presented in Chapter 6. Chapter 7 integrates these models and generalizes them for multiple resource types.

1.4.3 Contribution 2: CADENCE

CADENCE, the Control Architecture for the Dynamics of Embodied Natural Coordination and Engagement, is the interaction architecture developed in this thesis to control the robot's multimodal behavior.

Initially, CADENCE is used only for modeling turn-taking and action execution. Chapter 4 defines and motivates the use of timed Petri nets, the formalism on which CADENCE controllers are built. Chapters 5–7 describe successive iterations of CADENCE, each adding a new turn-taking capability.

Finally, CADENCE is expanded in Chapter 8 to encompass social cognition outside of turn-taking and action, such as natural language semantics and common ground, in order to achieve situated collaboration with a human.

1.4.4 Contribution 3: Autonomous interactions

We demonstrate autonomous interactions in four novel domains, which combine social interaction and manipulation on a humanoid robot. In Chapter 3, we implement the imitation game "Simon says" based on observations of a prior Wizard of Oz interaction in that domain. In Chapter 5, the human and the robot collaboratively build the Towers of Hanoi. In Chapter 6, the human and the robot engage in open-ended play about objects on a tabletop. In Chapter 9, the human and the robot collaboratively design and build a block model of a house.

1.4.5 Contribution 4: Evaluation of human-robot dynamics

In Chapters 5, 6, and 9, we also quantitatively evaluate the effects of each iteration of the system through user studies. We characterize the effects of system changes on task outcomes, fluency, and balance of control.

CHAPTER II

RELATED WORK

We start with ways in which the phenomenon of turn-taking has been studied in children (Section 2.1) and in adult communication (Section 2.2). For the rest of the chapter, we describe related computational systems.

2.1 Developmental psychology

The equalized management of shared resources during cooperation is a distinct characteristic of human social activity. Children divide resources and rewards equally at a very early age, even before they learn to count [120]. In contrast, even the most genetically related of the great apes are self-motivated, leading all of the rewards to be gained by the alpha ape after any collaborative work. A consequence of the alpha ape's unwillingness to reciprocate after being assisted is an inability to maintain cooperative activity. Unrewarded subordinates learn that there is no self-benefit to cooperation and refuse to assist the alpha ape in further trials, landing the team in a Pareto-suboptimal Nash equilibrium [112].

Research in developmental psychology has investigated preverbal interactions in infants with their mothers and has found that turn-taking emerges very early on in these mother-infant dyads. Tronick defined dyadic phases of social interaction consisting of initiation, mutual-orientation, greeting, play-dialogue (turn-taking), and disengagement [116]. Previously it was believed that all illusion of turn-taking was due to the mother's scaffolding and regulation of mutual gaze and response timing, but Trevarthen argued in [114] that the infant adapted to the mother's expressions as well, resulting in reciprocal play and communication.

In addition, behavioral analysis has been performed to understand the high synchrony in mother-infant interactions. Badalamenti et al. characterized mother and infant gaze-on, gaze-off, and gaze turn events as correlated stochastic processes [5]. Yu et al. used information-theoretic measures to quantify the synchrony of multimodal data recordings of mother-infant interactions [129].

Closely related to the timing of turn-taking interactions is the problem of contingency detection, which describes an agent's ability to detect perceptual changes that occur in response to the agent's own actions. Such changes could be environmental or behavioral, resulting in the learning of physical affordances or social behavior. In [78], Movellan implements a contingency detection controller using binary audio signals; the agent's action timing uses an information-maximizing approach inspired by infant vocalizations. Contingency detection has also been implemented using vision-based approaches [65]. In turn-taking, the ongoing perception and actuation of contingent events are the glue that binds together the entire reciprocal interaction and gives it its cyclic rhythm.

Because the innate skills for reciprocal interaction and cooperation are so deeply embedded in humans since infancy, they are easy to take for granted. Ideally, providing robots with a similar foundational capacity for reciprocity allows them to engage in social situations with humans in a way that is so intuitive for human users that they can take this behavior for granted as well.

2.2 Conversation analysis

Extensive treatment of turn-taking can be found in the linguistics literature as well, especially in the subfield of conversation analysis. A seminal piece is the work of Sacks et al. [94], which describes organizing principles for turn-taking in conversation. Key structures are the turn-constructional component, describing the syntactic units that

can form turns, and the turn-allocational component, describing how participants select subsequent speakers during discourse. [98] extends this work to describe practices employed by speakers to resolve overlapping speech. Feedback signals such as *uh-huh* or *okay*, known as backchannels, have also been analyzed for their role in structuring dialogue during joint activity [128, 7].

In contrast to the approach of analyzing turn-taking in terms of the linguistic content of the dialogue, other discourse analysts additionally address the role of nonverbal cues. [35] and [86] describe the presence and distributions of gaze shift, gesticulation, body motion, and paralinguistic cues such as prosody, drawls, and syllable-clipping in shaping conversation. Collectively, these cues form signals that convey intentions to yield, hold, seize, and avoid the floor. In addition, gaze and gestures play roles in progressively conveying semantic understanding so as to repair miscommunications as early as possible [30]. The diversity of these cues demonstrates the highly multimodal nature of the turn-taking process.

2.3 Spoken dialogue systems

Raux et al. developed a spoken turn-taking model based on a finite state machine [89], and later incorporated cost-based decision-making. Another model was based on bidding for turns with an importance score derived from the belief state and the expertise of the user [103]. Prior work in speech systems has also enabled users to barge in during certain dialogue states, and has focused on improving the accuracy of detecting such barge-ins [92, 108].

Some spoken dialogue systems have also aimed to capture more naturalistic turn-taking by focusing on the incremental nature of the problem. An early architecture developed by Allen et al. supporting incremental understanding and generation was applied to multiple domains, including information retrieval and customer service [2]. Schlangen and Skantze more recently presented a conceptual model for framing

architectures for incremental speech processing [101]. Traum et al. also showed an incremental speech processing system for multiparty interaction [113]. Incremental systems in a finite domain have been demonstrated to enable understanding sooner than the end of an utterance [34]. This allows the system or the user to interrupt the interaction partner with full understanding, a point we investigate more thoroughly in Chapter 3.

The spoken dialogue community offers many transferable results in terms of how to handle speech communication, but this is not a complete picture in a multimodal context. In situated collaboration, spoken turns may be coordinated with physical actions such as gesture and manipulation. Gaze is exchanged with the conversational floor, but is also used to attend to instrumental parts of the task. Turns, interruptions, and incremental understanding and generation are based on all modalities of behavior.

2.4 Embodied conversational agents

One of the earlier architectures for controlling multimodal behavior is BEAT [22], which was used to control virtual conversational characters that were extended by real-world perception of the human. The architecture was used to evaluate the effects of some turn-taking cues on the lifelikeness of the agents [23]. Certain turn-taking cues have been found to be effective when used by virtual agents, such as the control of eye gaze [50].

Bohus and Horvitz have also developed turn-taking models to be used for multiparty conversation with situated agents [12]. A more recent model uses a cost structure to make turn decisions [13]. The Ymir Turn Taking Model (YTTM) also supports multi-party turn-taking [111].

Work has also been done on predicting opportune moments for a virtual human to backchannel based on human corpus of head nods and speech [77]. Within a situated dialogue setting, data-driven approaches have been used to determine valid response

locations based on features from the human speech signal [73].

A recent architecture used for long-term engagement with an ECA, called DiscoRT [85], shares similar goals to CADENCE and is the latest version of a collaboration manager that implements the theory of SharedPlans [43]. Like CADENCE, DiscoRT is dedicated towards real-time collaborative discourse using social modalities, including gesture and gaze while speaking. In contrast to CADENCE, DiscoRT uses a model of turn-taking for utterances that assumes one atomic exchange at a time, and although tasks in DiscoRT are interruptible, turns are not.

2.5 Human-robot interaction

Research on turn-taking in social robots is motivated by similar goals to the research on ECAs or speech systems. For example, a conversational robot such as that demonstrated by Matsuyama et al. may regulate floor usage in a multiparty conversation [69]. However, some of the challenges of HRI differ from those of ECAs. Robots require time to move through physical space, and additionally, they must negotiate resources such as shared space and objects through turn-taking with human collaborators. Such bottlenecks arising due to embodiment are not an issue in virtual agent communication.

The work of [91] and [48] has addressed some of these challenges by identifying and generating multimodal "connection events" in order for a robot to maintain engagement with a human interaction partner. Other systems have also been developed to control multimodal dialogue for social robots. The work of [55] controls dynamic switching of behaviors in the speech and gesture modalities. The framework of [82] controls task-based dialogue with parallel modalities and supports task-level interruptions through explicit user commands, such as saying "stop."

In a vast number of social robot systems, turn-taking is an emergent behavior that arises from other interacting processes as opposed to being explicitly controlled.

A seminal example of such a robot is Kismet [16]. Kismet's control architecture did not represent or control the conversational floor, generating action only through the combination of an emotion regulation system with reactive behavior that gave rise to turn-taking. Similarly, Kose-Bagci et al. showed emergent turn-taking in a drumming interaction [57]. Our research aims to progress the field beyond emergent turn-taking because it does not account for breakdowns such as overlaps and extended silences, and it places disproportionate responsibility on the human to recover from such breakdowns.

Fluency has also been investigated in human-robot collaboration scenarios. Hoffman evaluated the effects of fluency when a robot generated anticipatory actions [47]. Another system enabled a robot to play an assistive role by handing construction pieces over using the human's eye gaze to anticipate intention [95]. In this context, the robot is subservient to the human's intentions. In a contrasting approach, Chaski is a task-level executive scheduler that assigns actions to both the human and the robot by minimizing idle time [104]. Schedulers are excellent for generating theoretically optimal plans, but a potential drawback is that reducing human control over forming task plans can lead to poorer mental models, engagement, and execution fluency. A goal of improving turn-taking is to achieve better balance of control in the interaction.

Other prior research focuses on subcomponents of the action coordination problem by studying individual modalities in a controlled fashion. For example, the work of [81] analyzes the function of gaze behaviors in designating speakers and auditors. [76] shows the design of gestures that communicate hesitation. [96] addresses the synchronization of communicative gesture with speech. Drawing from prior work in the field, our goal is to combine many such individual components into an integrated system, with the hope of iteratively increasing the communicative capabilities and overall social competence of a robot.

CHAPTER III

INFORMATION FLOW IN INTERACTION

From the related work presented in Chapter 2, we have many clues as to what socially appropriate behavior looks like in humans, and the precedents of a handful of systems that can accomplish social behavior. Our goal now is to develop a social interaction architecture built around the idea of a computational turn-taking model that is socially appropriate and applies across domains. The first challenge is a lack of available data in which robots behave in such a manner with humans. To build such a model, we must first understand how prevailing approaches cause robots to generate turn-taking in appropriate or inappropriate ways, then isolate the signals, patterns, and other interaction phenomena that will inform the development of improved approaches.

Later in this thesis, we alternate implementation of an autonomous system with the evaluation of it that informs the next iteration. To bootstrap this process, we start with an exploratory data collection representing the kind of multimodal social interaction that we are concerned with. This chapter describes the first and only Wizard of Oz experiment in this thesis; henceforth, all systems are autonomous, built upon the hypotheses of prior versions and evaluated in new domains.

In this data collection experiment, we collect and code data from 23 human subjects playing "Simon says" with the Simon robot. These results are the foundations from which much of the rest of this thesis is developed. Our key result from analyzing this data suggests that *minimum necessary information (MNI)* is a robust indicator for determining the human response delay to the robot across multiple phases in the interaction (Section 3.3.2). The data also show exclusions in the speech channel,

Figure 1: A participant plays "Simon says" with the Simon robot.

a precursor to the idea of managing *interaction resources* that we develop later in Chapters 6 and 7.

At the end of this chapter, we follow up this analysis with an autonomous version of the system in this domain. We discuss how this implementation leads to insights for further development and investigation.

3.1 A hypothetical turn-taking model

We start with a hypothetical computational model of how turn-taking can exist within a larger social interaction framework. This serves as the computational context in which we formulate the experiment described in Section 3.2.

3.1.1 Turn-taking as a Markov process

Due to the unobservability of other minds, human-human communication contains uncertainty and errors. Even with excellent perceptual capabilities, people still fall victim to unintended interruptions, overlapping speech, and awkward silences [98]. When moving to human-robot communication, the problems are intensified by noisy and limited sensor data. Given this combination of factors, it is natural to search for a

representation that is designed for modeling uncertainty, such as a dynamic Bayesian network, and view turn-taking as a problem of estimating which partner in the dyad has the floor to speak or act.

Such a model can be conceived as the first-order Markov process shown in Figure 2. At each time step t, both the robot (R_t) and the human (H_t) can be in one of four floor states: [Seizing, Passing, Holding, Listening]. Passing and seizing are the two transitory states where the floor is transitioning from one person to the other, while holding and listening are the two floor states of the dyad during a turn. Theoretically, R_t and H_t should always be in a seizing/passing or holding/listening configuration. But in reality many of the other "error" configurations will also have a non-zero probability. For example, at a pause in the dialog it is common to see both parties try to seize the floor, and then one decides to relinquish to the other. Or the listening party may try to seize the floor before the holding party makes any passing cues, commonly called a barge-in. The research challenge thus posed is to learn the parameters of this model from data, and involves two primary research questions:

- Timing model: This represents how and when the human and the robot transition from state to state, i.e., the human transition function $P(H_t|H_{t-1}, R_{t-1})$, and the robot transition function $P(R_t|R_{t-1}, H_{t-1})$.

- Observation model: The robot states are fully observable to the robot, but the robot has to infer the human's hidden floor state through sensory observations. $P(O_t|H_t)$ describes how the sensor data reflects the human floor state H_t.

The timing model describes the fundamental timing or structure of when people naturally take turns. The robot can use the timing model to determine if a person is being contingent or engaged, as well as decide when it might want to take a turn in order to avoid a collision. When perception is ambiguous, timing provides a feed-forward signal for the robot to keep on interacting. The observation model describes

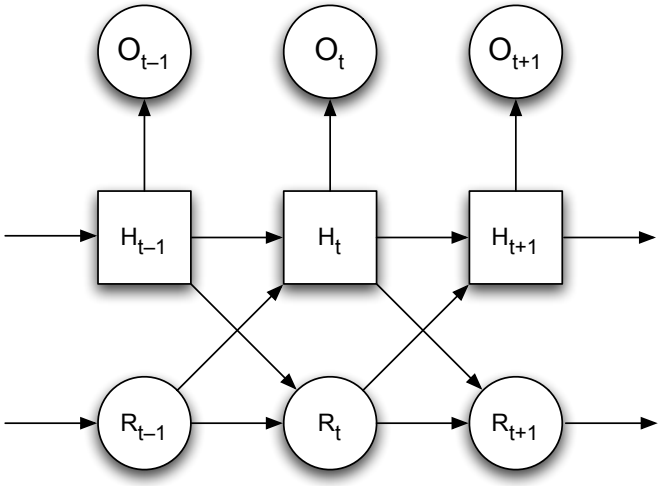


Figure 2: A theoretical model for human-robot turn dynamics, formulated as a first-order Markov process. At each time step both the robot (R_t) and the human (H_t) can be in one of four floor states: [Seizing, Passing, Holding, Listening].

the robot perception required to determine when people are about to seize or pass the floor, or when they are acting engaged. The observations form a feedback signal that keeps the overall model updated and allows the robot to understand what is currently transpiring in the interaction.

It is clear that the success of such a model depends in part on the availability and success of sensory observations that encapsulate important turn-taking signals, such as those described in [35, 86]. Then, timing parameters can be measured relative to these signals. Our approach is to analyze interaction data in order to find general assumptions that can be used to construct such a model. We thus conduct an experiment in which we collect a diverse selection of turn-taking episodes, both good and bad, through a combination of teleoperation and randomly generated timing variations. We then hand-code this data to learn about human-robot turn-taking behavior that can later be executed autonomously.

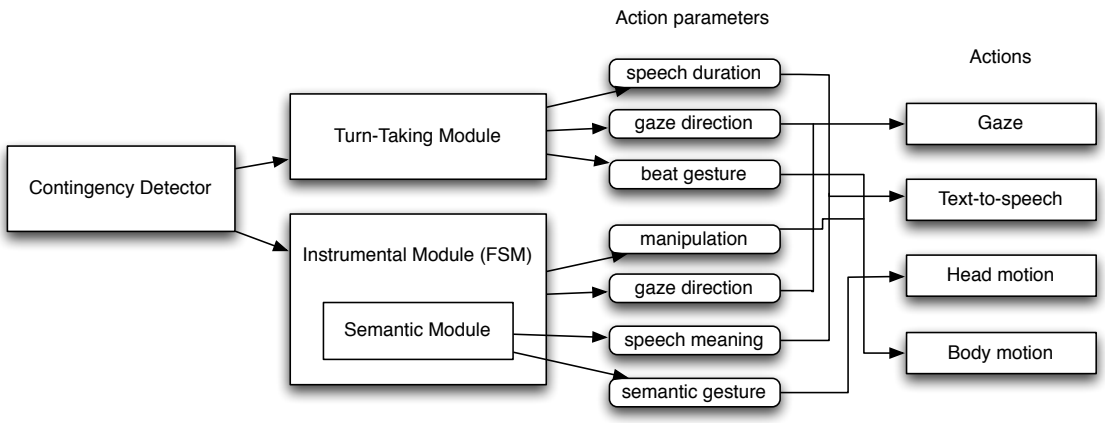


Figure 3: A proposed architecture for turn-taking. Parameters are specified by a context-free Turn-Taking Module and context-dependent Instrumental Module. Parameters from both modules are used to instantiate robot actions.

3.1.2 Context-free turn-taking

Figure 3 shows our current concept of an architecture for turn-taking. The architecture focuses on the specific channels of gaze, speech, and motion, which are independently well studied in HRI. Actions in these channels are parametrized, such that specific parameters can be decided by either the domain-specific Instrumental Module or the generic Turn-Taking Module in order to generate the final behavior.

The separation between the Instrumental Module and Turn-Taking Module highlights the principle dichotomy between domain-specific robot capabilities and context-free interaction behavior. That is, we hope to extract as much domain-independent turn-taking behavior as possible in order to create a transferable module. In this experiment, we focus on turn-taking in the specific domain of a "Simon says" game and present some analyses that lead us closer to this goal.

3.2 Experiment: Simon says

This next section describes a data collection experiment in which our robot is teleoperated to play "Simon says" with a human partner. The game is attractive as an

initial domain of investigation for its multimodality, interactive symmetry, and relative simplicity, being isolated from such complexities as object-based joint attention. We collected data from a total of 27 human subjects. For 4 subjects there was a problem that caused data loss with at least one logging component, so our analysis includes data from 23 subjects. We collected approximately 4 minutes of data from each participant.

3.2.1 Sensors

The sensors recorded were one of Simon's eye cameras, an external camera mounted on a tripod, a structured light depth sensor (Kinect) mounted on a tripod pointed at the human participant, and a microphone worn around the participant's neck. The computers used for logging data were synchronized to the same time server.

3.2.2 "Simon says" domain

The domain is an imitation game based on the traditional children's game "Simon says." Figure 1 shows the face-to-face setup. The game has a leading and a following role; the leader is referred to as "Simon." We divide the interaction into a game phase and a negotiation phase.

In the game phase, the leader can say, "Simon says, [perform an action]." The available actions are depicted in Figure 4. The follower should then imitate that action. The leader can also say, "[Perform an action]," after which the follower should do nothing, or else she loses the game. The leader concludes the set after observing an incorrect response by declaring, "You lose!" or "I win!"

In the negotiation phase, the follower can ask, "Can I play Simon?" or say, "I want to play Simon." The leader can then transfer the leadership role or reject the request. The leader also has the option of asking the follower, "Do you want to play Simon?" or saying to her, "You can play Simon now." The leader and follower can exchange roles at any time.

3.2.3 Robot behavior

All of the robot's behavior is organized into states in a finite state machine (FSM). The 15 states available to the teleoperator are described in Table 1. Each state in the FSM controls the robot's three channels of communication:

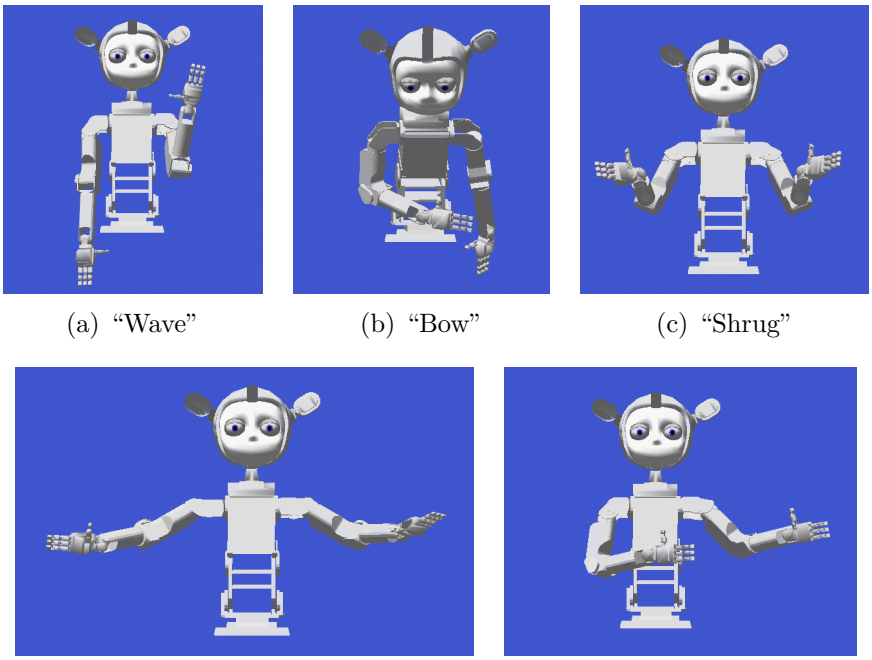
- *Body animation* – the actions of the game as shown in Figure 4. The speed of the animation was selected uniformly at random from a safe range.
- *Speech content* – an utterance randomly selected from the group of valid sentences for the state. Each state had 1-3 sentences as options.
- *Gaze direction* – gazing at the person's face using a visual servoing mechanism with the eye camera, or gazing away from the person.

To increase variation in the robot's executed behavior, the system injected delays randomly as follows. There was a 0.25 probability that the system would inject any delay at all; this was to ensure a basic number of responses that matched the teleoperator's intent. Then for each modality (speech, gaze, gesture) there were independent rolls for whether that modality would inject a delay at the beginning, the end, both, or neither. Each delay was uniformly sampled between 0–2 seconds.

The robot was teleoperated by myself using a keyboard interface to select specific FSM states. There was additionally an option of interrupting the current state, for a total of 16 keys. All of the keybinds were on one side of the keyboard to reduce the contribution of the keypress interface to the timing of the interaction.

3.2.4 Protocol

Participants were provided an explanation of the game and the available actions. They were not told that the robot was being teleoperated. The participants were told to adhere to a set of keywords when speaking to the robot. They were then given about a minute of practice with the robot to familiarize themselves with the



(a) "Wave" (b) "Bow" (c) "Shrug"

(d) "Fly like a bird" (e) "Play air guitar"

Figure 4: Actions in the "Simon says" game.

interaction and memorize the five actions. During this time they were allowed to ask clarifying questions to the experimenters. After the practice session, data collection commenced, and they were told to avoid interacting with the experimenters.

After the data collection was complete, subjects completed a survey about their experiences. The questions were similar to those in [23].

3.3 Results and analysis

Because our goal here is to understand human timing in turn-taking, our analysis focuses on human responses to the robot's different signals. We ask the questions: Which signal is the most reliable predictor of human timing? What is the timing model and distribution? This informs how a robot should shape its expectations about the timing of human responses, as well as emulate these parameters in order to produce human-like behavior. In this section, we present an analysis of experiment

Table 1: FSM states available to teleoperator.

State	Description
Hello	Start the interaction ("Hello, let's play Simon says").
Bye	End the interaction ("Thanks, that was fun").
Request	Request to play Simon ("Can I play Simon now?").
Accept	Accept request ("That's fine with me").
Deny	Deny request ("No, not yet").
Simon says	Select an action command starting with "Simon says."
Do this	Select an action command.
Win	Conclude the set by winning ("Ha ha, I win").
Lose	Admit to losing ("Oh no, I guess you win").
Can't do	Say "I can't do that."
Bow	Perform "bow" action as a follower.
Bird	Perform "bird" action as a follower.
Guitar	Perform "air guitar" action as a follower.
Shrug	Perform "shrug" action as a follower.
Wave	Perform "wave" action as a follower.

results about several components that contribute to the manifested timing of turn-taking.

3.3.1 Data coding

Figure 5 shows our interface for visualizing and coding the multimodal data. The data from the depth sensor, two cameras, and microphone can be played back in a synchronized fashion alongside an OpenGL visualization of the robot's joint angles

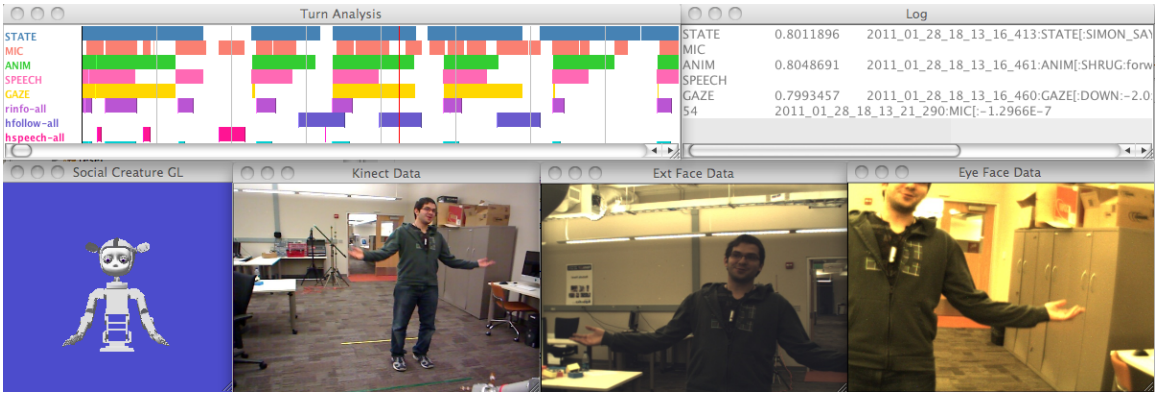


Figure 5: Interface for visualizing and video-coding the collected data.

and a live update of the text log of the robot behavior. The coders can scrub through the data and visually assess how their coded events align with other events in the data.

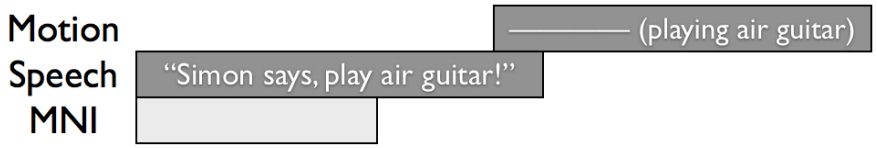
The specific data we now examine is the human *response delay*, which is the time between a referent event and the start of the coded human response. We separate the data collected from this experiment into game phase data and negotiation phase data, which show two different types of turn-taking interactions. All events that could not be automatically extracted from the robot's behavior logs were annotated independently by two coders, and for each event that was agreed upon, the coded time was averaged. The events were:

3.3.1.1 Game phase response

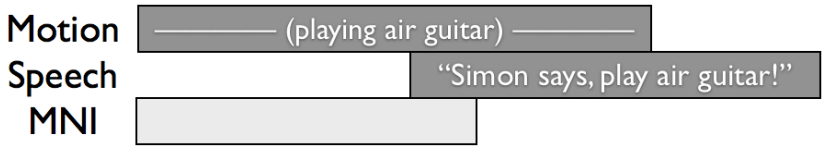
In the game phase data, the robot plays the leader and communicates using a mixture of speech, motion, and gaze. The human plays the follower and responds primarily with a motion, which is sometimes secondarily accompanied by a speech backchannel. For a more controlled data set, the game phase data includes only correct human responses to the robot's "Simon says" turns. The coder agreement was 100% for game phase events, and the average difference in coded time was 123 milliseconds.

3.3.1.2 Negotiation phase response

The negotiation phase response was also a human event. In the negotiation phase, the exchanges are shorter, and the robot uses speech but not any body animations to communicate. Most robot utterances are also too short for the robot to have time to gaze away and back to the human, so the robot primarily gazes at the human. The coder agreement was 94.2% for negotiation phase events, and the average difference in coded time was 368 milliseconds.



(a) All informative speech occurs before the animation starts.



(b) The action is conveyed through motion before the human knows whether or not to execute it.

Figure 6: Examples of coding robot MNI in the game phase.

3.3.1.3 Minimum necessary information (MNI)

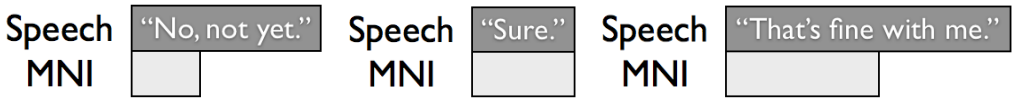
The minimum necessary information was a robot signal. This describes an interval during which the robot conveys the minimum amount of information needed for the human to respond in a semantically appropriate way. More explanation and reasoning for this signal is provided next in Section 3.3.2. Figures 6 and 7 show examples of MNI video coding. In the game phase, the human needs to know whether or not to respond as well the motion with which to respond, so the information end is the earliest point at which both of these are conveyed. In the negotiation phase, the information is usually marked by a pronoun. The coder agreement was 99.8% for robot MNI events, and the average difference in coded time was 202 milliseconds.

3.3.2 Minimum necessary information (MNI)

In order to characterize a predictive human response delay distribution, one needs to determine a reliable referent event. For example, some channel-based referent events are: the end of robot motion, the end of robot speech, or the moment when the robot gazes at the human after looking away. Histograms of response delays with respect to these referent events are shown in Figure 8 for both interaction phases. It becomes immediately apparent that not all of these signals are useful predictors. Specifically,

(a) Pronouns demarcate information for turn exchange.

(b) The emotive phrase announces the end of a set.



(c) Examples of acknowledgments.

Figure 7: Examples of coding robot MNI in the negotiation phase.

a good referent event should yield distributions that have these properties:

1. *Nonnegativity* – If the response delay is negative, then this referent event cannot be the cause of the response.

2. *Low variance* – The distribution should have low variability to allow for more accurate prediction.

3. *Generality* – The distribution should be consistent across different types of interactions.

Responses to the motion event and the gaze event both violate nonnegativity (Figure 8). Gaze has been demonstrated to be an excellent indicator in multiparty conversation domains [81, 12], but it is less predictive in this particular dyadic interaction; we suspect that it might show greater impact in a dyadic object manipulation task. The best channel-based referent event is speech, but 41% of human responses still occur before the robot finishes speech in the game phase.

We thus argue for a concept called *minimum necessary information (MNI)* — the minimum amount of information needed to be conveyed by the robot for the human to respond in a semantically appropriate way (that is, discounting barge-ins

or simultaneous starts). The best referent event to use is the end of the MNI signal. The response delay distributions to MNI endings are shown superimposed with the other distributions in Figure 8 and also fit to curves in Figure 9. MNI satisfies nonnegativity for both interaction phases and is relatively general. The means in Figure 9 are also within half a second from that of the distribution in [78]. We think this could be attributed to the higher processing requirement for the multimodal information content of this game.

3.3.3 Channel exclusion

We also hypothesize that human turn-taking follows conventions for managing exclusions per channel. We observed that although subjects did not wait for the robot to finish speaking before they moved, they usually waited for the robot to finish speaking before they spoke. This accounted for the differences in the distributions of response delays to speech shown in Figure 8. For responses to speech, the negotiation phase distributions were shifted in the positive direction as compared to the game phase distributions.

Additionally, we observed that people tended to avoid simultaneous speaking after a simultaneous start. There were 23 instances of simultaneous speech in the data set, spread across 10 subjects. Of these, 7 (30%) constituted backchannel feedback. The remaining 16 instances were simultaneous starts. Of the simultaneous starts, 3 resulted in the teleoperator interrupting the robot speech, 8 resulted in the human interrupting his own speech, and 3 resulted in a decrease in the human's speech volume. Although this is sparse data, this tendency to back off from simultaneous starts shows an adherence to channel exclusion.

This channel exclusion also has an effect on the response delay distributions to MNI. Compared to the game phase distribution, the negotiation phase distribution is slightly delayed due to this lock. However, the MNI is still relatively robust overall

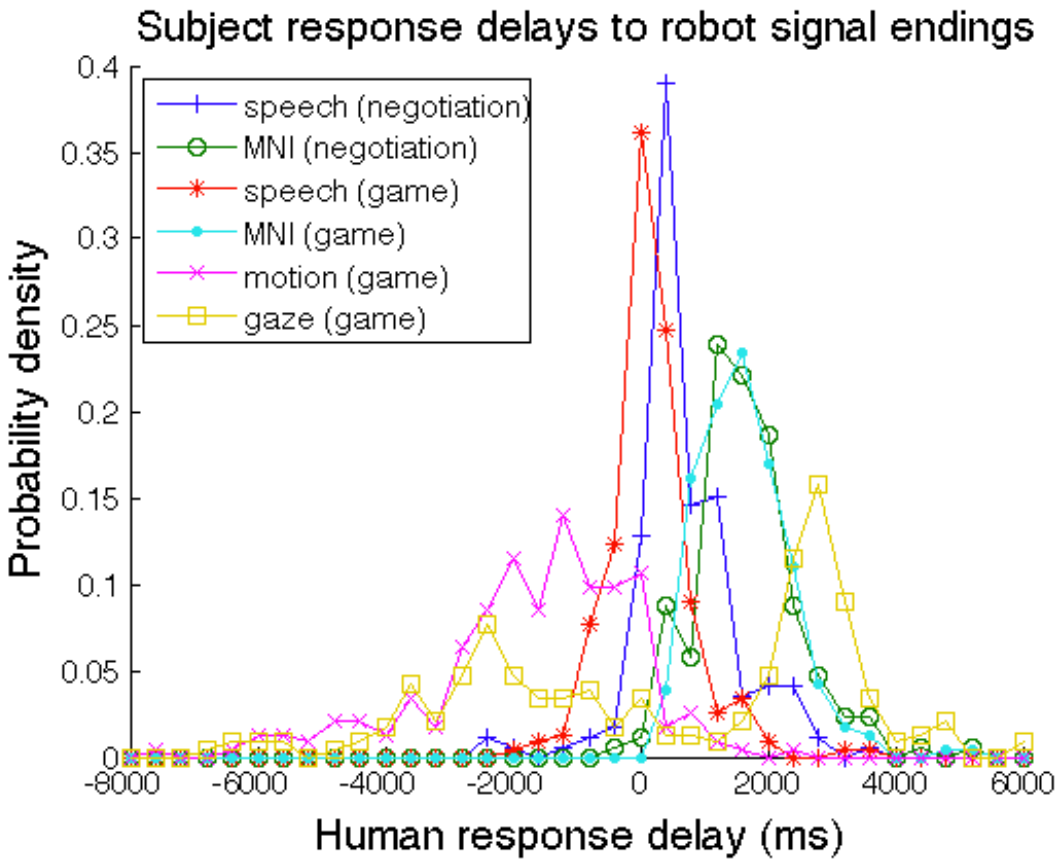


Figure 8: Histograms of human response delays with respect to all potential robot referent signals. Negative delays indicate that subjects responded before the robot completed its turn-taking action within that channel.

because the robot's speech contained a balance of shorter and longer utterances.

This domain had only one channel with a "lock," which was speech. One could envision a domain where there were exclusions in the motion channel. Both parties could need to move in the same space or need to use the same tool. These factors could lead to delayed responses. In addition, more or fewer exclusions in any channel could arise due to differences in cultural communication or personality.

3.3.4 Efficiency vs. adaptation

Turn-taking is a dynamic process, and timing can evolve as the interaction progresses. If we believe that MNI endings are stable referent events, we can use response delays to them to investigate how human responses change over time.

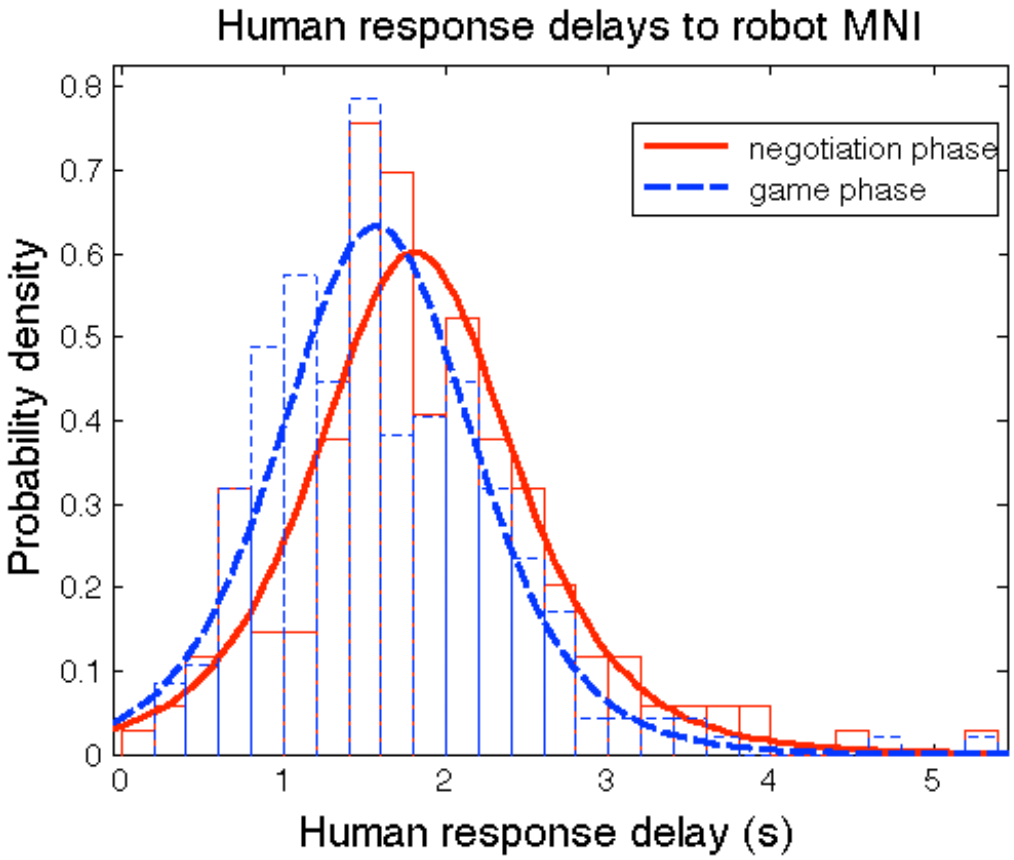
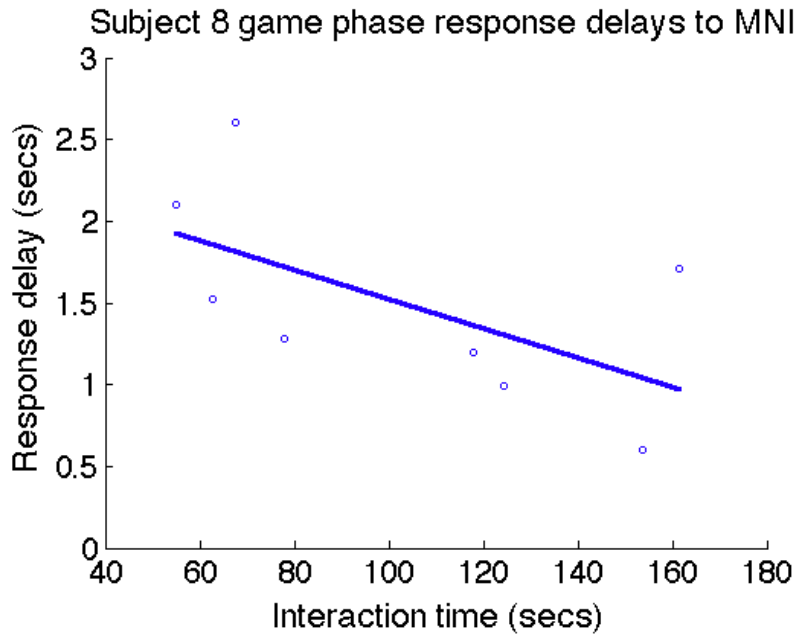


Figure 9: The delays of human responses with respect to robot MNI endings in the negotiation and game phases. The curves represent maximum likelihood fits to Student's t probability density functions.

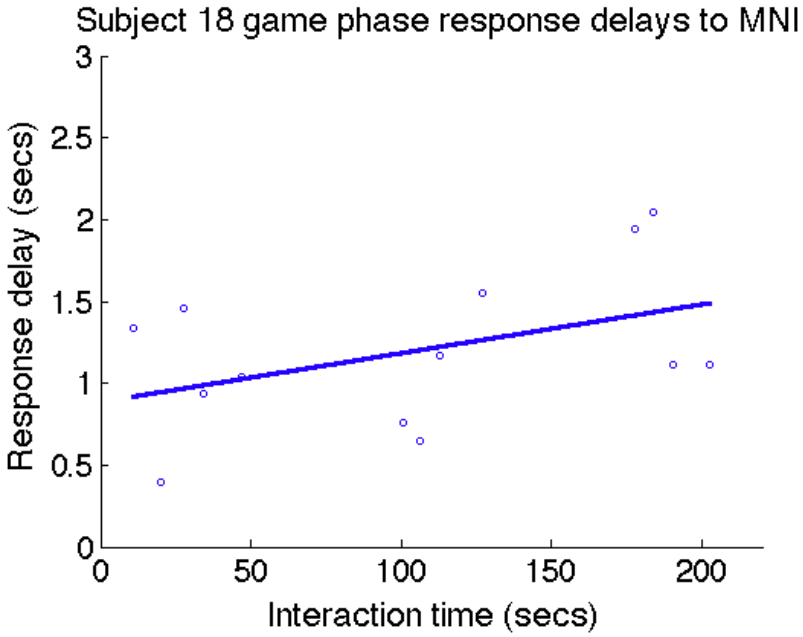
One phenomenon we observed in the data was the notion of increasing efficiency or fluency, as described extensively in [47]. We can characterize a response's efficiency as the inverse of the response delay after the MNI end — the lower the response delay, the higher the efficiency. For some subjects, their time to react decreased with practice, as less information was needed from the robot to react, and the response delays showed a downward trend. An example is shown in Figure 10(a). Nine subjects (39%) exhibited this trend in their data.

Although this interaction was too short to see a significant difference, we think that a robot can expect this change in any domain involving repetitive behavior that leads to improvement in performance. Leiser observed in [66] that repeated information exchanges between humans cause abbreviations in language due to decreasing information requirements, which suggests that responses would approach MNI endings with repetition. A well-practiced human-robot dyad may operate at a periodicity close to the MNI point, with plenty of overlapping in any channel that did not present an exclusion.

We hypothesize that there is also another phenomenon of adaptation, where one party can adapt to and gradually approach the other party's timing. We observed that certain subjects started to imitate the robot's mannerisms of speech and motion and actually slowed down their timing to be more similar to the robot's. An example is shown in Figure 10(b). Seven subjects (30%) showed this trend. With a robot behavior control system that was sensitive to turn-taking timing, this could occur in both directions, with both parties converging on a timing between their prior distributions.



(a) Efficiency – Subject 8 responds more quickly after more practice with the game.



(b) Adaptation – Subject 18 responds more slowly over time, adapting to the robot's behavior.

Figure 10: Changes in interaction timing.

3.4 Discussion

3.4.1 Limitations of context-free behavior

We consider the validity of the previously presented hypothetical model in light of these results. With respect to the timing model, we observed in this experiment that timing and behavior vary greatly across subjects. This makes it difficult to aggregate data across people, since the variance could overwhelm the duration of short turns. For a robot that works out of the box, another approach is likely needed, which is the focus of the rest of this thesis. However, it is still plausible that such a model could be trained over time as the interaction progresses, factoring in such effects as adaptation and increased efficiency with practice in order to adapt to the user's timing.

The notion of an observation model trained from sensor data is also problematic. Such a model could be trained by video-coding important turn-taking events on the sensor stream. However, in practice, many of the indicators are very subtle, such as eye gaze shifts, aspirations, mouth openings, and user-specific gestures. Such cues would be very difficult to detect accurately. The results also do not support the idea that modality signals themselves are reliable predictors of turn-taking behavior.

Most critically, both the timing and observation models depend on the concept of MNI, which significantly detracts from the notion of a completely context-free turn-taking model based only on exterior behavioral cues. In addition, determining the MNI may not be easy for less structured domains. For a "Simon says" game with only a handful of actions and utterances, it can be pre-coded easily for autonomous behavior, but this coding may be too arduous for other tasks. The iteration of CADENCE described in Chapter 8 addresses the significant complexities involved in automatically controlling turn-taking from semantic content.

For now, we position the principle of MNI as a useful way of understanding the turn-taking process, even in cases when it is not a convenient signal to come by. MNI conceptualizes turn-taking as a process of *information exchange*, in which the

production of surface behavior is dominated by intention and understanding and secondarily affected by markers for turn negotiation. One such example is the manner in which humans use "uh" and "um" followed by a pause when they are uncertain about what they are going to say, a behavior that signals delayed information transfer [31]. The turn state, timing, and observation signal will all be dominated by content in the subsequent turn, and only slightly modulated by the prefix of "um." The robot would do well to monitor its own information transmission in similar ways, which we address in later chapters.

3.4.2 Importance of interruptions

Given that people respond to the robot using the MNI point, we would expect awkward floor passing to result when the robot does not make use of this signal in its own floor passing behavior. Qualitatively, we observed that this happened several times in our experiment in both phases. In the game phase, this typically happened when the robot was the leader and continued to perform its gesture (e.g., playing air guitar) for too long after the human partner had already interpreted the gesture and completed the appropriate response turn. In many cases, subjects had to wait for the robot to finish once they were done with their response gestures. This also happened in cases where the human lost the game. We see examples where they notice the gesture the robot is doing, start doing the same gesture in response, then visibly/audibly notice they were instead supposed to remain still. In inappropriate segments, the robot still takes the time to finish its gesture before declaring that the person has lost. These examples illustrate the inefficiencies that the robot introduces when it does not have appropriate floor relinquishing behavior.

In the negotiation phase, the robot's awkward floor relinquishing behavior results in dominance over the human partner rather than just inefficiency. As mentioned previously, in this phase the turns are primarily speech-based. Thus, simultaneous

speech is a common occurrence. For example, after a pause both the robot and the human might start asking "Can I play Simon now?" This state of both parties trying to seize the floor is typical in human communication, and results in one of the parties relinquishing to the other (e.g., "Go ahead" or "Sorry, you were saying?"). However, in our experiment the robot's autonomous behavior was to keep going to the end of its turn, resulting in the human always being the party to back off. The few good examples of robot floor relinquishing were a result of the teleoperator acting fast enough to interrupt the robot's behavior and let the human have the floor.

The principle of MNI thus suggests that the robot should relinquish the floor earlier if the human has clearly conveyed understanding, rather than always insisting on completing the current action in its current state. Combined with the need to recover from simultaneous starts, we now have two strong arguments for the importance of smooth action interruptions in a turn-taking architecture. This leads directly into our next step: developing an autonomous version of this interaction that supports such interruptions.

3.5 Autonomous floor yielding

In this section we describe the modification to the robot's FSM implementation that enables autonomous floor yielding. We demonstrate the results with both a gesture-based and speech-based example.

Our implementation of autonomous floor yielding is achieved by allowing transitions to interrupt states in the FSM. Figure 11 depicts the difference between the former state machine and the new implementation. Previously, a state had to run to completion before the transitions out of that state started being evaluated. We achieve floor yielding by allowing some state transitions to start being evaluated, and optionally interrupt the state, prior to the state's completion. If the transition does not fire early, however, the behavior is the same as a normal state machine. The

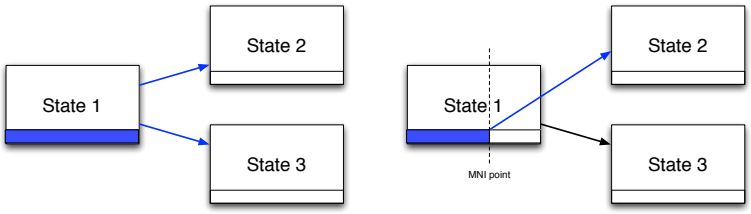
interaction designer specifies whether or not any given state can be interrupted or any given transition can fire early.

The interruption of a state signals that the current state should terminate as soon as possible so that the next state can activate. This means for example that any active text-to-speech process is destroyed, and the current robot joint positions are quickly interpolated to the starting positions of the next state. The timing of an interruption should potentially also affect the robot's behavior. A robot may want to treat an interruption that occurs after the MNI point of the state as a normal part of information exchange and proceed with the domain-specific activity, whereas an interruption that occurs prior to the MNI point could indicate a simultaneous start from which the robot should back off or attempt to recover.

This FSM modification allows us to set up a "Simon says" FSM very similar to the one we used in the experiment described earlier, but with the new feature that the robot can now have the expectation that after the MNI point in a particular speech or gesture being executed, there is the possibility that a transition to the next state will happen prior to action completion.

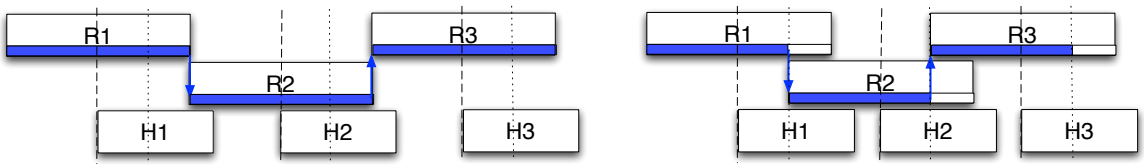
Figure 12 compares the timing of the FSMs with and without interrupts side by side to show how interrupts at the MNI point increase interaction efficiency. When the robot plays the leader in the "Simon says" domain, the robot autonomously recognizes the human's game actions using incremental gesture recognition on the Kinect sensor data and transitions as soon an action is recognizable (i.e. before the human finishes her action by putting down her arms). Compared to the original behavior, where the robot's actions run to completion, the overall interaction becomes more efficient and timing becomes dictated more by the human responses with less human wait time.

Another useful application of interruptions is resolving simultaneous starts. Using voice activity detection on the human's microphone, the robot can recognize that the human is in the process of speaking prior to recognizing the full sentence from the



(a) (b)

Figure 11: This illustrates a slice of an FSM; the bars at the bottom of each state indicate the progress toward the completion of that state's actions. Figure (a) represents a typical FSM, where a state finishes executing and then evaluates to decide on a transition. Figure (b) represents our interruptible FSM implementation, to achieve floor relinquishing. A state transition has the option of evaluating prior to the completion of the previous state, and based on this can interrupt the current action to proceed to the next state without completing the first.



(a) Turn-taking without state interruptions (b) Turn-taking with state interruptions

Figure 12: R indicates a robot turn, and H indicates a human turn. The dashed lines show robot turn MNI points and dotted lines show human turn MNI points. Figure (a) shows a state machine without interruptions; even when the human MNI point passes, the robot continues to complete the state's actions. Figure (b) shows how transitions that interrupt the current state can make the interaction more efficient.

grammar. If the human asks the robot, "Can I be Simon now?" at the same time that the robot says something, the autonomous controller interrupts the robot's current speech to gain a clearer signal of the human's speech for speech recognition. Compared to the original behavior, the robot is now able to allow the human to barge in rather than always requiring that the human back off during simultaneous speech.[1]

[1] Video demonstrations of both of these examples can be found at http://www.cc.gatech.edu/social-machines/video/interrupt-gesture.mov and http://www.cc.gatech.edu/social-machines/video/interrupt-speech.mov.

3.6 Architectural implications

With the aim of creating a general-purpose architecture for HRI, the computational model in Section 3.1 could be integrated with the interruptible FSM according to Figure 13. This architecture focuses on the specific channels of gaze, speech, and motion, which are independently well studied in HRI. Actions in these channels are parametrized, such that specific parameters can be decided by either the domain-specific Instrumental Module or the generic Turn-Taking Module in order to generate the final behavior. The separation between the Instrumental Module and Turn-Taking Module highlights the principle dichotomy between domain-specific robot capabilities and context-free interaction behavior.

In reality, the boundary between turn-taking and domain semantics is not so pronounced. The turn-taking model would need to give floor state estimation, which drives the domain-specific FSM, but that FSM also needs to tell the turn-taking model about the flow of information in each context, which will usually be highly semantic-based information. Then collectively they contribute parameters for robot actions.

Notably, the modules in Figure 13 are of completely disparate representations. This makes it more difficult to define precisely how they should interact, or characterize the system as a whole. In fact, it is possible to model all of them together using the same underlying formalism: the timed Petri net (TPN). The next chapter details this formalism and its application to multimodal interaction. All subsequent work in thesis then uses TPN modeling to build an integrated version of such an architecture.

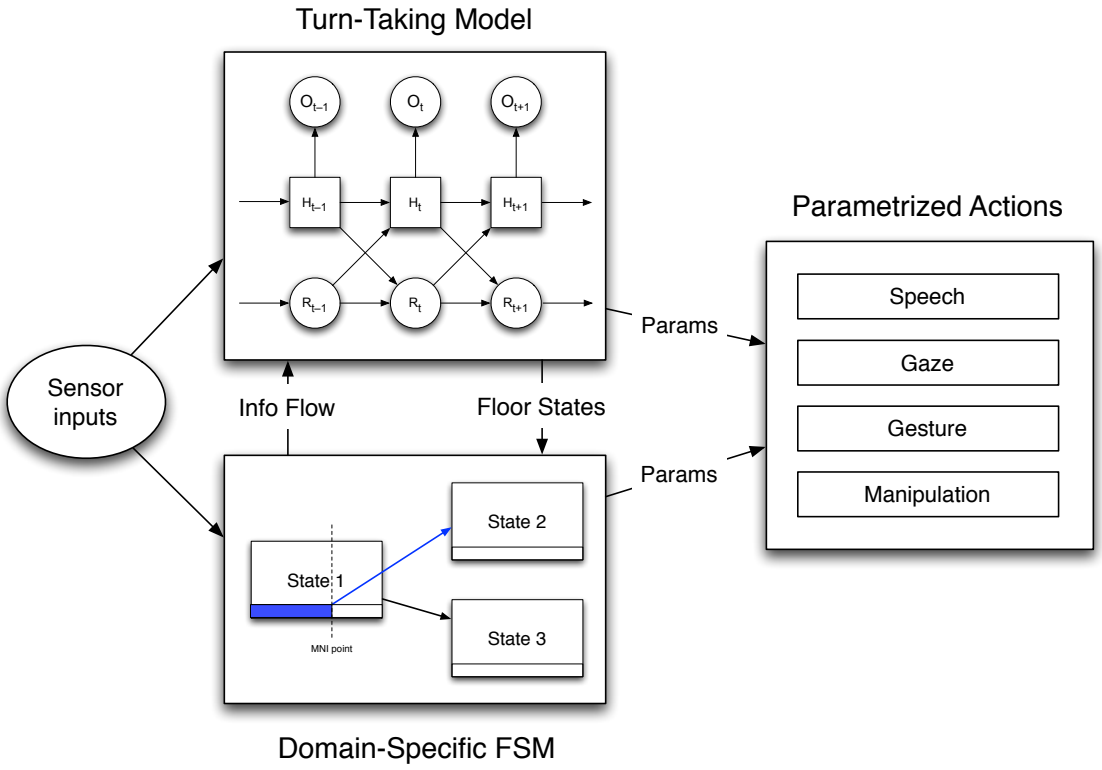


Figure 13: A framework for turn-taking in HRI: The turn-taking model tracks floor state estimation, which drives the domain-specific FSM. The FSM provides the turn-taking model feedback about the flow of information in the domain. They collectively contribute parameters for robot actions.

CHAPTER IV

TIMED PETRI NETS FOR BEHAVIOR CONTROL

The rest of this thesis describes the development of CADENCE, the Control Architecture for the Dynamics of Embodied Natural Coordination of Engagement. CADENCE includes an action architecture based on timed Petri nets (TPNs), an extension of Petri nets with additional modeling of timing. This decision is based on the modeling power of TPNs for combinations of temporally extended actions. We define this formalism and offer some intuition for its application to multimodal social interaction. We also argue for its advantages in scalability, generalizability, and representation of time. Implementation of this formalism are then used to build the systems in Chapters 5–7.

4.1 Requirements of social interaction

Cooperation between humans is characterized by temporally extended action. Humans perform communicative acts that engage the multiple modalities of speech, gesture, and gaze, while simultaneously applying their bodies to the tasks at hand. Such social exchanges often feature:

- *synchronization* over bottlenecks, as when taking turns with the speaking floor in a dialogue, handling shared objects in the environment, or waiting for the visual attention of a partner;
- *concurrency* of actions across participants, as well as of actions across the modalities of a single participant;
- *sequences* of conditions that must be met, as when following a plan or executing conversation following a situational interaction "script";

- *timing* management, so that the collaboration is efficient and that interaction dynamics meet expectations.

A robot or embodied agent designed to interact with a human in such a cooperative setting requires a behavior system that can model these types of interactions. We believe that timed Petri nets (TPNs) offer a natural representation for multimodal interactions, despite currently being relatively uncommon in robotics. Although almost any robot architecture can be molded to fit any small-enough problem, TPNs offer some representational advantages for scaling complex implementations and for transferring behavior between domains. We posit that timed Petri nets are a natural computational representation given the system requirements of social interaction scenarios.

First, the capability of multimodal action inherently requires action parallelism; an architecture in which only a single action is executed at a time would result in unnatural behavior, such as an arm gesturing action blocking a gaze action with the head. Simultaneous modality execution in such an architecture requires all cross-modal actions to be separately specified. Continuing the previous example, the ability to execute an action to say hello, versus wave, versus coordinating both would require three separate actions instead of just one action per modality. This poses scalability issues combinatorial with the number of resources and modalities. Petri nets, on the other hand, express synchronization and concurrency precisely and compactly.

In addition, in natural interaction, action modalities are not bijective with resource types. If this were the case, simpler automata like finite state machines (FSMs) could simply be operated in parallel on a per-modality basis. Humans use cues cross-modally as signals for turn-taking over the speaking floor [35]; an example is using gesture or eye gaze to yield the floor or suppress auditor seizing attempts. Robots also share resources between action modalities, such as gaze and head gestures, or arm gestures and manipulation.

Natural turn-taking in dialogue is also rife with starts and stops. In practice, turns are not taken in accordance with formal rules as in a chess game (or even as in the simplest systematics of Sacks [94]), but incrementally with frequent hesitations and interruptions [30, 98]. Such actions serve to convey uncertainty or intentions towards shared resources. Interruption handling is well supported in Petri net models.

Finally, we consider the importance of timing as a variable in many interaction decisions. Turn-taking is a function of action times, reaction times, and impatience. Markovian graphical models are intended for modeling memoryless probability rather than time, and, while applicable to many modern robotics problems, pose difficulties in capturing the nuances of real-time interaction where time variables do not always follow a geometric distribution [18]. We favor the distributed development of TPNs, in which timing factors can be decomposed into component subproblems represented as smaller TPN processes. These subprocesses can be learned or designed, then connected and synchronized. The TPN framework is well suited for expressing a distributed synchronized discrete event system such as multimodal turn-taking.

A central issue for progressing a robot's HRI capabilities is the problem of creating general "social intelligence" interaction modules that produce transferable behavior across multiple domains. In particular, our research is focused on the development of a general-purpose turn-taking module. This stands in contrast with the idea of discovering transferable *principles* through user studies, to be applied case-by-case in a hard-coded fashion. Our goal is to move away from a model where each new domain requires painstaking setting of each gesture and glance, and towards a model where new domains only specify new semantics and a few behavioral parameters. We offer a systems perspective on the advantages and disadvantages of using TPNs for multimodal interaction modeling as compared to some of the more commonly established representations in robotics.

4.2 Background

The literature on Petri nets is highly diverse and includes many variants, each with their own approaches to typing and timing. Petri nets have been popular in previous decades for workflow modeling due to their rich representation and intuitive graphical notation. They have historically been used for modeling, but have not been as commonly used for control. Previous work on robot control using Petri nets include applications in assembly [130] and manufacturing [21]. More recently, they have been used as supervisors for multi-robot control in a robot soccer domain [8, 62]. Holroyd has also used Petri nets for the realization of Behavior Markup Language (BML) [49].

In modern robotics, Markov decision processes [54] and Markov chains are extremely popular representations for sequential decision-making. Finite state machines (FSMs) are also more commonly used. We think that in most cases where multiple FSMs are parallelized, Petri nets would probably be better suited. We also think that Markov models, which are meant to model uncertainty over sequences, are awkward for modeling temporally extended actions in HRI. These issues will be addressed in Section 4.5.

4.3 Formal definition

The formalism we detail here integrates various Petri net modeling techniques that specifically support the control of multimodal reciprocal human-robot interactions. We find that Petri nets, and specifically TPNs, are an intuitive and elegant representation for developing autonomous controllers for HRI turn-taking scenarios. In the remainder of this section we describe the formalism and its application to turn-taking control.

4.3.0.1 Petri nets

A basic Petri net, or Place/Transition (P/T) net, is a bipartite multigraph comprising two finite disjoint sets of nodes, places and transitions. A multiset of directed arcs connects the node types in an alternating fashion. Places are used to represent robot state and can contain a natural number of tokens; control is transferred through token movement throughout the graph. The mapping of places to tokens is called a marking $M : P \to \mathbb{N}$, and is the Petri net's implicit state representation. More formally, a Petri net is a 5-tuple $N = (P, T, I, O, M_0)$, where:

- P is a finite set of places,
- T is a finite set of transitions, where $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$,
- $I : P \times T$ is the input function directing incoming arcs to transitions from places,
- $O : T \times P$ is the output function directing outgoing arcs from transitions to places, and
- M_0 is an initial marking.

Places contain a nonnegative integer number of tokens, which can represent any kind of resource. This could be a queue of parameters to be processed, a discrete variable, or a shared tool. In general, the state of a Petri net is implicitly represented through the marking $M : P \to \mathbb{N}$, the number of tokens in each place. In our system, tokens are typed objects mapped onto a value; this variant is sometimes referred to as a colored Petri net. Figure 16 shows examples of tokens having the types of Animation, SpeechAct, and Vec3.

In addition to the above general Petri net semantics, the firing mechanics for our system are as follows:

- A token $(k : \sigma) \to (v : \sigma)$ is parametrized by type σ and has value v of that type.

- A place $(p : \sigma) = \{k_1 : \sigma, k_2 : \sigma, ..., k_z : \sigma\}$ is parametrized by type σ, and contains a list of same-typed tokens.

- A transition $t = \{\mathcal{G}(I), \mathcal{F}(M, I, O)\}$ is controlled by a guard function $\mathcal{G}(I)$ and a firing function $\mathcal{F}(M, I, O)$.

- A guard function $\mathcal{G}(I) \rightarrow \{0, 1\}$ is an indicator function describing the logic for enabling transition t as a function of the inputs of t. An enabled transition executes the firing function until the transition is no longer enabled.

- A firing function $\mathcal{F}(M, I, O) \rightarrow M'$ takes as input the current graph marking M and produces new marking M' by removing tokens from the inputs of t and assigning tokens to any of its outputs, following type rules. The transition is considered to fire whenever a new marking is produced.

The firing function runs until the guard function is no longer satisfied, which requires the transition t to alter the graph marking in a way that changes the guard function inputs. This results in the transition disabling. A transition can induce such marking changes by transferring ownership of tokens between places, by destroying unneeded tokens in input places, or by spawning new tokens.

In our system, places and tokens are also typed; this variant is known as colored Petri nets. The control logic for guard functions depends only on the presence and absence of tokens in places, but the firing logic for a transition can unpack the typed data contained within tokens to perform an operation.

Typical guard functions in the system are AND-logic and OR-logic expressions, but any boolean expression is possible. Our system also uses the common addition of inhibitor input arcs, which allow places with tokens to prevent transitions from enabling.

Petri nets have a specific visualization scheme in which places are drawn as circles, transitions as rectangles, directed arcs as arrows, and tokens as small filled circles

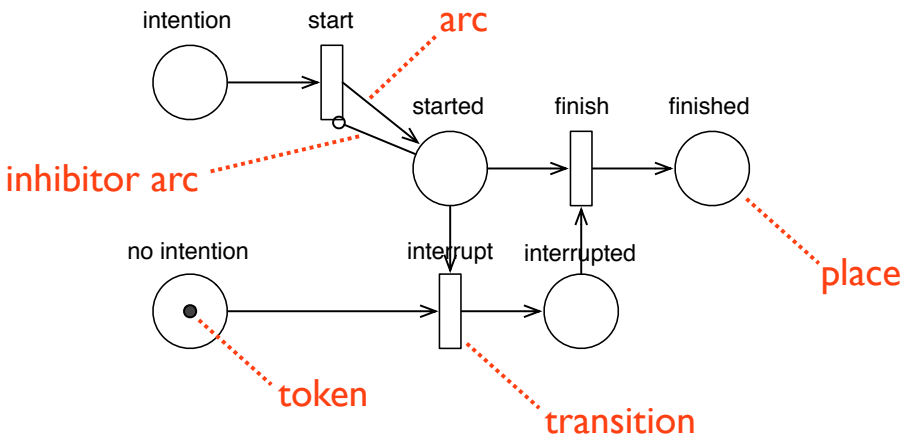


Figure 14: An interruptible action template (to be revisited in Chapter 5), with Petri net primitives labeled.

inside of places. Inhibitor arcs are drawn with a circular endpoint. We have labeled these graph primitives for the reader in Figure 14. More details on standard Petri nets and their applications can also be found in a detailed survey by Murata [79].

4.3.1 Timing

Timing control and analysis is made possible with the following additional components. For a more extensive overview of the different kinds of TPNs, see [118].

- The system clock $C(i, \tau) \rightarrow \tau'$ determines how the current time τ updates to the new time τ' at each cycle i.
- A transition $t \stackrel{+}{=} \{\delta_e(), \delta_f(I)\}$ is additionally associated with an enabling delay function and a firing delay function.
- An enabling delay function $\delta_e() \rightarrow d_e$ calculates the delay d_e before the transition is enabled from the time that the guard function evaluates to true.
- A firing delay function $\delta_f(I) \rightarrow d_f$ calculates the expected delay d_f after the time when the transition is enabled but before the transition fires.

The clock module determines the rate at which the system is executed, the needs of which may vary depending on the application. A clock that updates τ faster than real-time is useful for simulations. The delay functions in our system are varied in structure and include immediate timers, deterministic timers, and stochastic timers (e.g. following a Gaussian distribution). An example application of injecting a timer into a control sequence is waiting in a system state for a certain duration before proceeding, such as adding a 1-second delay when a human bottlenecks the robot before a verbal request to address the bottleneck. Shared restrictions to all time distributions in a Petri net result in different classes of Petri nets, such as stochastic Petri nets (exponential) or time Petri nets (deterministic intervals) [118].

Timing history is tracked in a distributed manner in our system by associating histories of timing intervals $[\tau_\alpha, \tau_\beta)$ with certain graph primitives, defined as follows:

- For a place p, τ_α is recorded when $|p|_i = 0 \rightarrow |p|_{i+1} > 0$, and τ_β is recorded when $|p|_i > 0 \rightarrow |p|_{i+1} = 0$. These intervals are segments of time during which the place owns tokens.

- For a transition t, τ_α is recorded when $\mathcal{G}(I)_i = 0 \rightarrow \mathcal{G}(I)_{i+1} = 1$, and τ_β is recorded when $\mathcal{G}(I)_i = 1 \rightarrow \mathcal{G}(I)_{i+1} = 0$. These intervals are segments of time during which the transition is enabled (and thus executing the firing function).

- For a token k, given that $k \in p_i$ at cycle i, τ_α and τ_β are recorded when $p_i \neq p_{i+1}$. Tokens can be owned by nil. These intervals are segments of time describing how long the token has been owned by any given place.

Such historical data can be useful for making certain turn-taking decisions. For example, one can decide whether to act based on the amount of time spent acting previously. In Chapter 5, this data is used to simulate the user parameter of initiative. Although decisions based on timing history are non-Markovian and can be difficult

to analyze closed-form using currently available mathematical techniques, simulation can be used to analyze systems of arbitrary complexity.

For the purposes of controlling robot behavior, we sometimes design transitions that are intended to fire continually for an extended duration (i.e., while engaged with a human or throughout an entire experiment); this fits within the semantics of TPNs but contrasts somewhat with traditional Petri net modeling, in which incoming tokens to a transition are intended to be consumed immediately.

4.3.2 Discrete events

There are various ways to represent the types of discrete events that occur in a multimodal dialogue. From a transcript logged from system execution, one can produce a bar visualization depicting the alignment of the events. This format can be helpful for annotating, in conjunction with data playback, and could look something like Figure 15.

In the figure, the beginning and end of each segment represents a discrete event that is important to the interaction — that is, a state change that potentially affects a system decision. Hence, what we are concerned with is the specification of a Discrete Event Dynamic System (DEDS), which describes the potentially concurrent or asynchronous alignments of important event chains throughout a system execution. A DEDS can be expressed as a Petri net, which provides a useful set of semantics shared for both control and analysis. A more detailed review of Petri nets can be found in [79].

4.3.3 Application to multimodal interaction

Figure 16 depicts a simplified example of how multimodal state can be represented using TPNs. The tokens are numbered and labeled with their values. In our system visualization, we also show filled and partially filled transition rectangles, which communicate firing delay expectations. In the example, the robot is about three

quarters of the way through executing a "wave" animation, has just finished executing text-to-speech of the phrase "Hello," and is about to start looking at the human partner.

Figure 17 depicts a hypothetical base set of dependencies between all such modules in our system, which would be implemented as Petri net subgraphs. The edges in the diagram indicate that modules interface by connecting Petri net primitives. When the graph is extended to support new domains, additional dependencies between modules may be introduced. The context model needs to be instantiated on a per-domain basis, as it selects semantic actions to be handled by the behavioral layer; these actions are timed by the turn-taking module. Our approach to this division within the interactional layer is further described in [110]. We emphasize that the loosely denoted layers do not imply a strict order of execution, as a Petri net represents a distributed event system.

4.4 Relation to common alternatives

In this section we discuss the relationship between Petri nets and two common state-based representations, finite state machines (FSMs) and Markov chains.

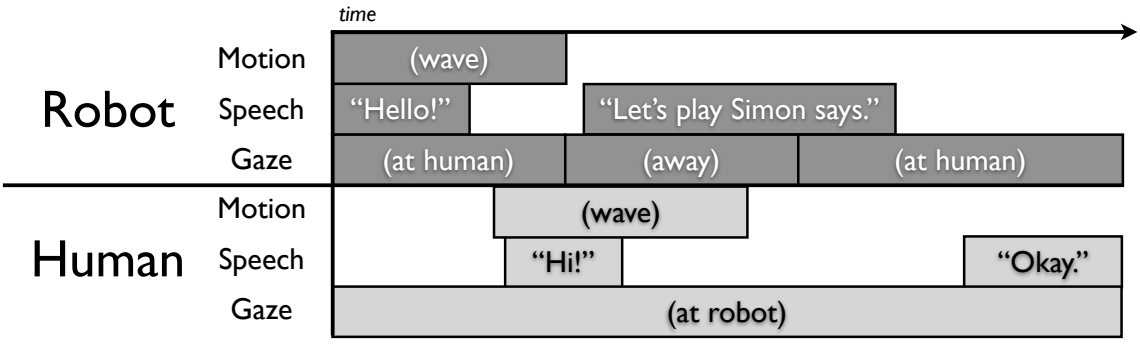


Figure 15: An example of event alignment for a multimodal interaction.

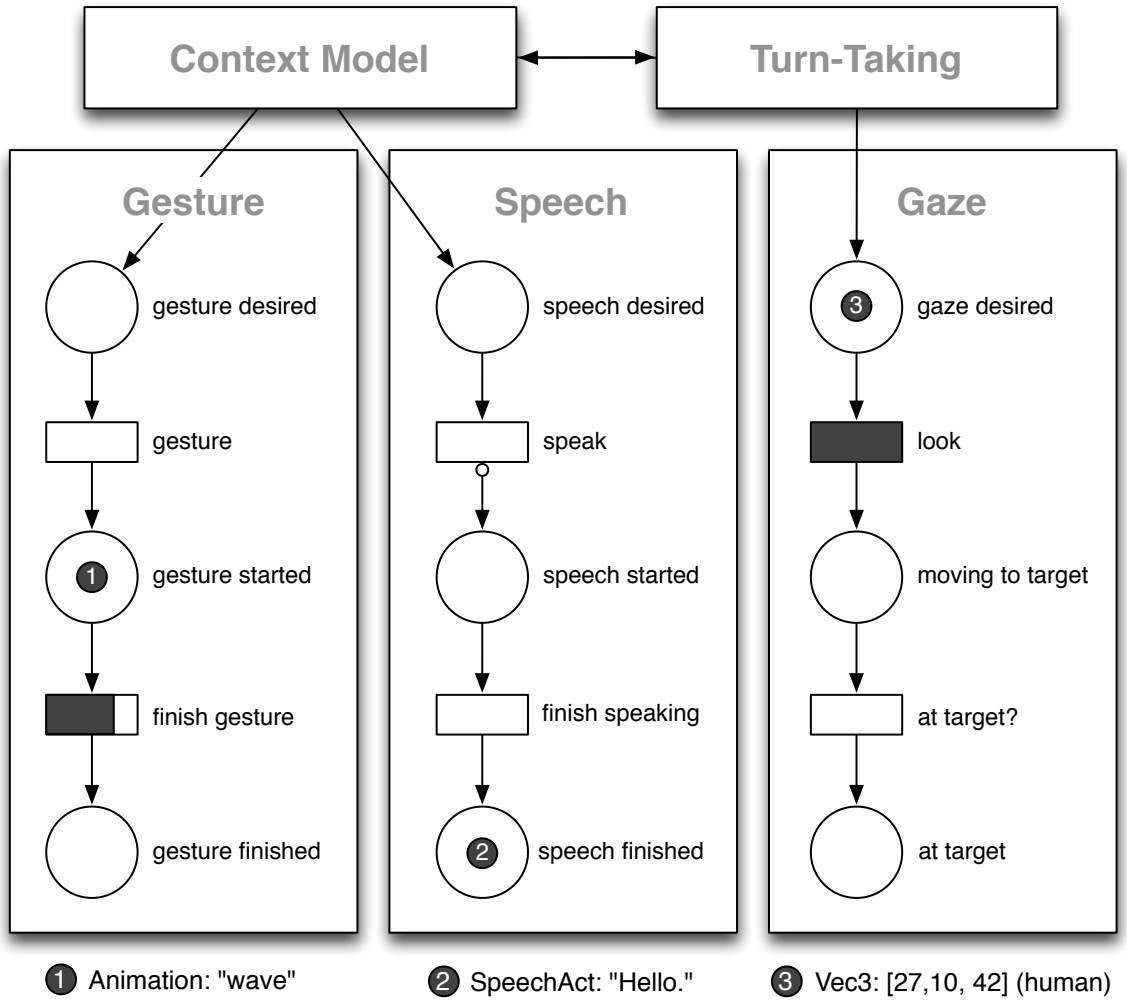


Figure 16: A simplified example of how multimodal state is represented in a Petri net.

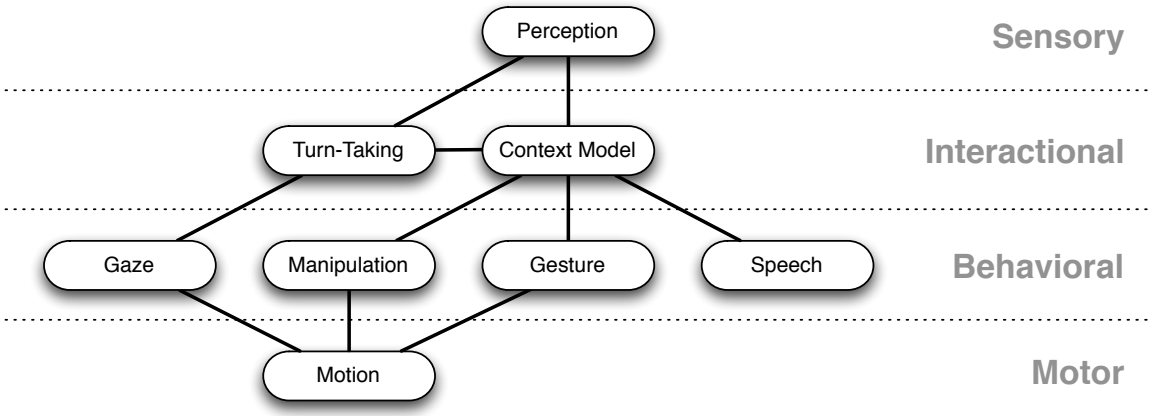
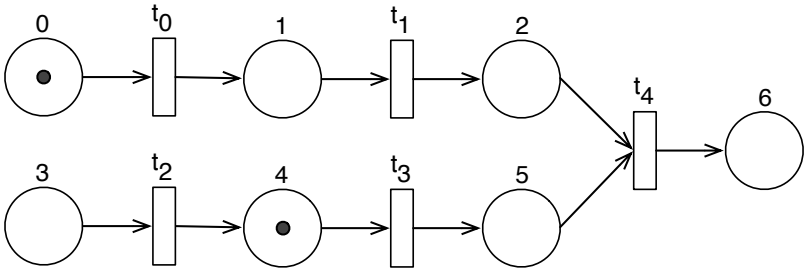
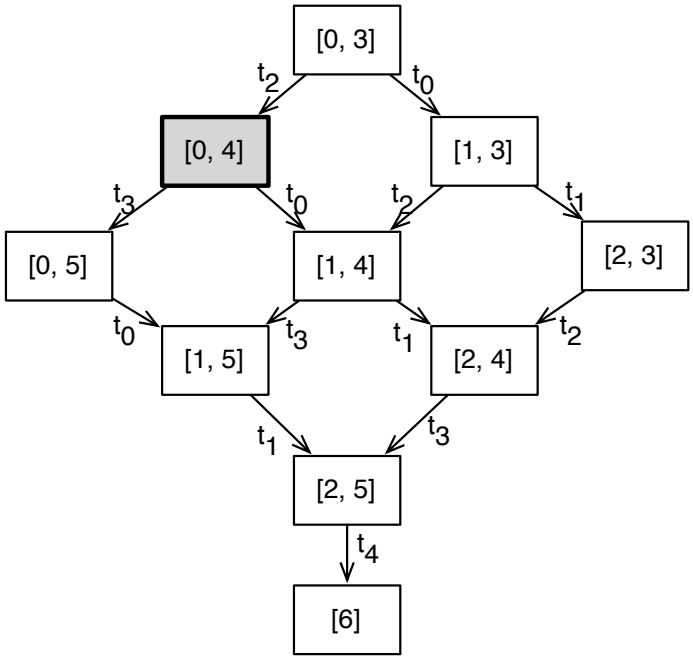


Figure 17: Base dependencies of modules in an initial version of the architecture.



(a) Example Petri net that models sequences, concurrency, and synchronization.



(b) Finite state machine (and reachability graph) corresponding to the Petri net in Figure 18(a).

Figure 18: A Petri net with its corresponding finite state machine (FSM). The FSM states are written as tuples of Petri net places that concurrently contain tokens.

4.4.1 Finite state machines

One of the most commonly used automata for agent action is the finite state machine. Finite state machines are well suited for modeling simple sequential control. They are formally defined as:

- S is a finite set of states,
- X is an alphabet of input symbols,
- Y is an alphabet of output symbols,
- $T : S \times X \rightarrow S$ is a transition function between states,
- $O : S \times X \rightarrow Y$ is an output function, and
- $s_0 \in S$ is an initial state.

Both Petri nets and FSMs match a development style that focuses on modeling conditional rules. Structurally, any FSM can be represented in Petri net form. Such a Petri net requires transitions to have a maximum of one input place and one output place, and the net contains exactly one token. The key limitation of the FSM is that only one state is active at a time, which is what limits its utility in applications requiring concurrency. It is difficult for systems relying on FSMs to model both synchronization and concurrency. One can parallelize FSMs to represent concurrency, but then synchronization across the FSMs is not modeled explicitly, which results in unpredictable behavior and scalability issues—oft-cited caveats of the subsumption architecture [17]. If one wants to model synchronization instead, the crossproduct of any concurrent conditions must be taken to enumerate the full state space. An example is shown in Figure 18(b), the FSM for the Petri net in Figure 18(a); this is also the reachability graph of the Petri net, which connects markings that are successively reachable from each other as a result of transition firings. For those

already familiar with FSMs, Petri nets offer a principled way to integrate multiple FSMs.

4.4.2 Markov chains

A Markov process describes a sequence of states obeying the Markov property, meaning that the probability distribution for transitioning from one state to the next is conditionally independent of those at previous or future time steps. A discrete-time Markov chain is defined as:

- S is a finite or countable set of states,
- $X(n) = X_0, X_1, ...$ is a sequence of S-valued random variables at time steps $n = 0, 1, ...$
- $Pr(X_n = j | X_{n-1} = i)$ for $i, j \in S$ is a transition probability function obeying the Markov property.

Like FSMs, Markov models are state-based representations. Figure 18(b) can depict a Markov chain with the following adjustments: each directed edge is labeled with a probability of transitioning, and each state also has an edge directed at itself for self-transitions. The transition function is applied at each time step to yield a posterior distribution indicating the likelihoods of being in each state. Certain classes of TPNs and Markov chains have equivalent dynamics. For example, stochastic Petri nets have exponential firing times associated with their transitions and thus can be mapped to continuous-time Markov chains.

Markov models are state-based, so they suffer from the same representational issues as FSMs. It also turns out that the addition of sequential probability is not particularly well suited for timing in interaction. These issues are discussed next in Section 4.5. Other Markov models, like hidden Markov models (HMMs) or partially observable Markov decision processes (POMDPs), have similar characteristics to the

Markov chain. In Section 4.5, we refer frequently to Markov chains, but the representational characteristics being discussed generally apply to all types of Markov models.

4.5 Issues in modeling

Here we summarize some of the relevant concerns about developing systems to control robots in HRI settings.

4.5.1 Scalability

A Markov chain with N states can be represented by an $N \times N$ transition matrix containing transition probabilities between pairs of states, usually trained from data in an unsupervised fashion (or potentially hand-coded, which can be a less than intuitive process). Because all potentially concurrent conditions must be coupled together to form states in Markov chains and FSMs, this representation poses an issue for the system's scalability. The state space grows exponentially with the number of concurrent conditions because all values possible for each dimension must be combined with all other dimensions' values. This can be problematic for systems that model many modalities.

The enumeration of the minimal set of combinations of concurrent conditions may be unintuitive. One strategy easily enabled by a Markov chain representation is to allow combinations of any conditions' values regardless of the actual relationship between them. The true structure of whether conditions co-occur is reflected in the sparsity of the trained transition matrix. This approach can be beneficial for smaller problems but naturally worsens the model's scalability. A way to deal with the tractability of state-based representations is to expend effort on pruning and merging states after training, a familiar practice for users of POMDPs. In comparison, the number of Petri net nodes scales linearly with additional concurrent conditions (adding additional places), thereby remaining much more manageable and compact

in the face of increasing complexity.

4.5.2 Generalizability

Another issue with state-based representations is that it is currently difficult to generalize the resulting model to new domains (although transfer learning is an active research area). Because concurrent conditions must be coupled together to form states, this necessarily includes domain-specific conditions in all of the states. Any modifications to the state space require that the model be retrained, whether it be changing domains or extending behavior. Also, there's no portion of a Markov model that can be easily extracted and reused; the system stands as a whole or not at all.

In contrast, it is possible for a TPN to be decomposed into multiple subprocesses, each a TPN in itself, that are connected by interfaces. Properly abstracted subprocesses can be reused by connecting them to new domain-specific models. Multiple people could develop separate subprocesses with an agreed-upon interface, as in standard software engineering. The graphical notation facilitates the communication and interpretation of such designs.

TPNs are also easy to modify locally. When new nodes are added, connected transitions must have their firing dynamics specified (i.e. probabilistic firing distributions, interval timers, etc.), but the entire system does not need to be retrained. The combination of the modularity of TPN process design and its relative extensibility makes it an attractive representation for iteratively developing the social cognition of a robot or agent.

4.5.3 Time representation

Another drawback of Markov chain-based representations for this application is the indirect representation of time. In Markov chains, time passed in a state is implicitly represented through self-transition probabilities. That is, when a state in a discrete-time Markov chain transitions to itself repeatedly over discrete time steps before

it switches to another state, some multiple of the discrete time step passes as a result. The time passed thus follows a geometric distribution. The continuous-time Markov chain describes a variant with time following the exponential distribution, the continuous analogue of the geometric distribution.

In the particular application of modeling the timing of events within a temporally extended action, it is indirect to work in probability space instead of time space. It is also restricting to limit oneself to the memoryless exponential and geometric distributions, which can result in inappropriately timed transitions. Some techniques overcome this by including discretized time as a dimension in the state space. An example is [18], which used a time-indexed POMDP to train an agent's turn-taking within a driving domain. Incorporating time in this way aggravates the scalability issue by exploding the state space and forces a tradeoff between tractability and model expressivity. Again, much effort needs to be expended on pruning or merging states.

In addition, memoryless probabilistic state switching can result in nonsensical behavior, as there is a chance that states will switch unnaturally quickly. Although Markov models can generate summary time characteristics such as expected times or durations, the model does not represent any temporal continuity. In order to use the model reasonably and safely, some additional filtering or self-transition bias is likely required [119], which disrupts the dynamics that the system was trained to model. The TPN allows the setup of temporally coherent actions, where transition firing dynamics can be specified intuitively in terms of the relevant space (time), whether they are derived statistically from data or hand-coded.

4.5.4 Analyzability

A system's analyzability describes the ease with which one can answer questions about its dynamics. Examples are the percentage of time spent in a state, or the probability of one condition given another. For example, in analyzing human-robot teamwork, it

may be important to evaluate the percentage of time the human or robot was idle.

When it comes to formal methods for analysis, Markov chains have the advantage. They have been popular because they are backed by efficient algorithms that can answer questions about uncertainty regarding past and future observations. In fact, Petri nets are often converted to their Markov chain duals for performing analysis. However, this superiority applies specifically to standard memoryless models, i.e. following geometric or exponential distributions. Realistically in HRI, such assumptions don't apply to timing requirements of interactions. Semi-Markov models can be used to model other timing distributions but are much more difficult to analyze because the efficient algorithms, e.g. Baum-Welch, no longer apply.

In our work, we have found that simulation-based methods of analysis are required to be maximally general [26]. Although many runs are needed and results are not as satisfying as from closed-form proofs, the method allows for mixtures of transitions with arbitrary firing timing. It seems to us that significant complexity of naturalistic interactions must be sacrificed in order to model them as Markov chains, with their limited scalability and representational accuracy, for the sake of closed-form analyzability. When the original problem requires so much simplification for the sake of tractability, it actually makes little sense to point to "optimal" policies and inferences in POMDPs and Markov chains. We thus consider that TPNs still offer an attractive alternative over the throwaway system designs fostered by state-based representations.

4.6 Discussion

Our perspective is that state-based representations such as FSMs and Markov chains can work quite well if one only needs to deliver an interaction within a single domain. The indirectness of the representation of time in the Markov chain can make development more difficult, but there are workarounds possible.

Where representational issues pose more of a concern is in developing general systems that support increased complexity. The literature on HRI contains many principles on how robots should act, as ascertained through user studies featuring single behaviors. Still new principles are being discovered and published all the time. It is less than ideal to have to hand-code such principles into the robot's behavior for each new domain of interaction with no hope for transfer. If such principles could be encapsulated in modular, transferable processes, perhaps better progress could be made towards achieving richer social behavior in interactive robots. Considering the limitations of the prevailing methods for producing robot actions, TPNs seem to offer advantages in scalability, generalizability, and representation of time.

4.7 Summary

Multimodal human-robot interactions tend to feature synchronization, concurrency, condition sequences, and timing requirements. Timed Petri nets are well suited for modeling these types of control flows. Due to representational advantages when scaling to more complex systems, they are worth considering as an alternative to the more commonly used methods of FSMs and Markov chains.

CHAPTER V

INTERRUPTIBLE BEHAVIOR

In this chapter, we describe the first incarnation of CADENCE based on the timed Petri net formalism described in Chapter 4. This architecture is designed dually for the control *and* analysis of timing in multimodal reciprocal interactions. *Reciprocal* describes the robot's human-centered social tendency, including motivation to engage users and maintain balanced turn-taking behaviors. This chapter focuses on the development and evaluation of a robot interrupting its own behavior in order to yield resources to the human.

A system that attempts to model significant amounts of concurrent and interruptible behavior can be complex. When such a system needs to scale to incorporate novel behaviors or additional modules, it can be difficult to evaluate how various factors and their combinations contribute to overall interaction dynamics. We thus leverage TPN simulation in order to provide such factored characterizations of the system. In Section 5.2.4, we describe how TPN simulation can be used as a technique to analyze HRI systems dynamics.

To demonstrate the benefits of our system in more detail, we also conduct a focused study of the effects of a particular system extension, action interruptions, on turn-taking dynamics. We describe the semantics and implementation of such action interruptions in Section 5.1. We describe our evaluation methodologies for the extension in Section 5.2, which includes a traditional between-groups user study with 16 human subjects and a simulation experiment with 200 simulated users. By analyzing the simulation data in conjunction with the user study data, we discuss in Section 6.4 how we are able to achieve better understanding of the effect of action

interruptions on system and interaction dynamics.

5.1 Action interruptions

In this section, we describe what we hypothesize is one fundamental skill for robots to interact fluently with humans: the ability to interrupt temporally-extended actions at arbitrary times. We motivate this design principle in Section 5.1.1 and describe its implementation details for our system in Section 5.1.2 utilizing the formalism introduced in Section 6.1. Sections 5.2 and 6.4 then focus on evaluating this addition.

5.1.1 Motivation

One contributor to fluency in humans is the equalized management of shared resources during cooperation — a distinct characteristic of human social activity [120]. When two humans share a bowl of popcorn or hold doors open for each other, they engage in seamless turn-taking of shared spaces. The conversational "floor" is another important shared resource. When humans converse to exchange information, they yield the floor when appropriate [98, 35]. In the presence of shared resources, this continuous give-and-take process is necessary to balance control between two partners and thus maximize their contributions to the joint activity.

We believe that robots can achieve higher interaction fluency by using an action execution scheme that dynamically yields shared resources and control to humans in order to maintain efficient and reciprocal turn-taking. Humans should be able to exert fine-grained control over robots with the same kinds of subtle mechanisms used to influence other humans. On an extreme level, humans do have ultimate authority over robots through the emergency-stop button, but such coarse levels of control are not helpful for accomplishing cooperative tasks.

Here, we investigate how a robot can effectively yield control of two specific resources — the speaking floor and shared space — in the form of speech and manipulation action interruptions. We hypothesize that managing shared resources in this

way leads to improved interaction balance and thus better task performance.

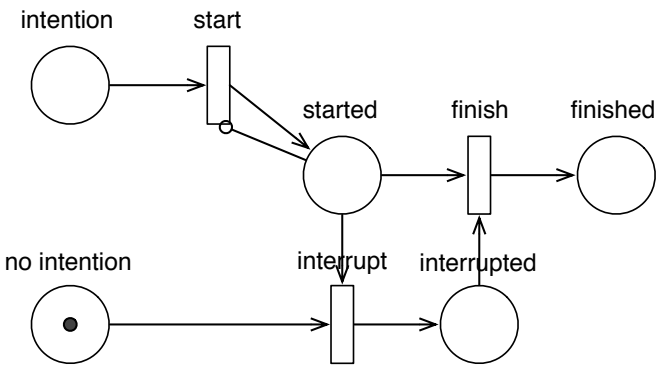
5.1.2 Implementation

5.1.2.1 Action Atomicity in Reciprocal Interaction

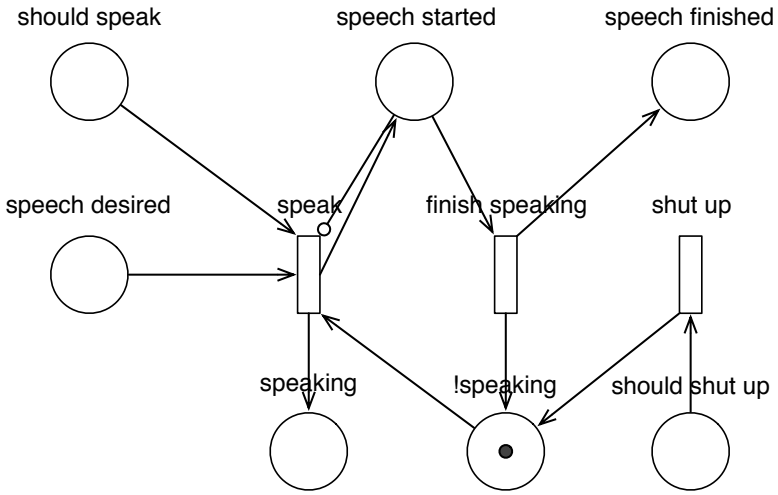
Traditionally in HRI scenarios, basic-level actions such as gestures, gaze directions, and speech commands are triggered in response to stimuli and then executed to completion. For example, detecting a human reactively produces an action — an emblematic gesture of "wave," or a text-to-speech greeting such as "Hello, how are you?" One limitation of always completing communicative acts, as we discovered in previous work [25], is that it potentially adds superfluous bottlenecks to the interaction. It is also misleading regarding the internal state of the robot. Continuing to speak when the message is already across implies that the speaker believes the listener has not received the message.

Other systems have endeavored to address this topic. The work in [89] and [83] are examples of dialogue systems in which speech interruptions in particular are supported. Interruption has also been addressed more indirectly through an approach of behavior switching [55]. We believe that interruption should be explicitly handled as its own intention ("stop this" rather than "react to something higher-priority"). In our implementation, we also support interruptions through multiple modalities of behavior. And our model is expressed formally in order to control complex system states precisely and facilitate further analysis. Often in ad hoc behavior-switching architectures, the modules interact in unpredictable ways that lead to "emergent" behavior that is difficult to understand or recreate.

The general idea is that an action should be divided into key stages. An atomic action should not be to "wave," but instead to "start waving," "keep waving," and "stop waving" at arbitrary points in time depending on the task context. You might continue waving at a friend across the street until she catches your eye, after which you can stop immediately, because the necessary information has most likely been



(a) An example template for a fluent action.



(b) A speech process reflecting the structure of the template.

Figure 19: Visualization of behavioral actions, which are subgraphs of the Petri net behavior system.

transmitted. This shift in action atomicity seems necessary to achieve fluency. Existing interactions with robots do not break down constantly, as some amount of fluency emerges naturally from simple reactive behaviors [57], but they require a human to adapt his timing to the robot and back off when needed. Because the robot's behavior is not reciprocal, the robot tends to dominate control of interaction timing. This may be acceptable in certain situations, but should occur purposefully within a larger process of joint intentionality, not as an accidental side effect of the robot's action formulation.

5.1.2.2 Template for Fluent Actions

Figure 19(a) shows a generic template for a fluent action in the form of a Petri net subgraph. A separate process decides whether the agent should have the intention of performing the action; if so, it deposits a token in $p_{intention}$ and destroys the one in $p_{no-intention}$. This token triggers t_{start}, which deposits a token in $p_{started}$. After the action has started, if an external process deposits a token in $p_{no-intention}$, this in combination with $p_{started}$ triggers $t_{interrupt}$, which deposits a token in $p_{interrupted}$. From there, t_{finish} can trigger and deposit a token in $p_{finished}$. If the action is not interrupted, the action terminates normally through t_{finish} without ever encountering the interrupt loop. External processes can attach transitions to $p_{finished}$ to read the token values, essentially subscribing to the message for the event finishing.

The action atomicity paradigm previously described is clearly manifested in this action template through the three separate transitions of t_{start}, $t_{interrupt}$, and t_{finish}.

Figure 19(b) shows an example behavior in the system, the speech process. The subgraph is similar in structure to the basic action template. One difference is the additional contextual parameter of $p_{speech-desired}$ describing the particular speech act to execute, which combines with the intention of $p_{should-speak}$ in order to trigger t_{speak}. Another is the presence of the additional variables of $p_{speaking}$ and $p_{!speaking}$, which contrasts with the notion of a speech act that has started or finished (because a speech act may contain pauses in speech between strings of utterances).

A goal of this line of research is to develop an inferential turn-taking model within our architecture, which requires defining an interface between context models and generic interaction behavior in terms of information flow. For example, a turn-taking model may decide whether the robot should or should not speak in the speech process. If the robot has information to pass, it can produce that utterance; otherwise, it can opt for a backchannel.

5.2 Experiments: Towers of Hanoi

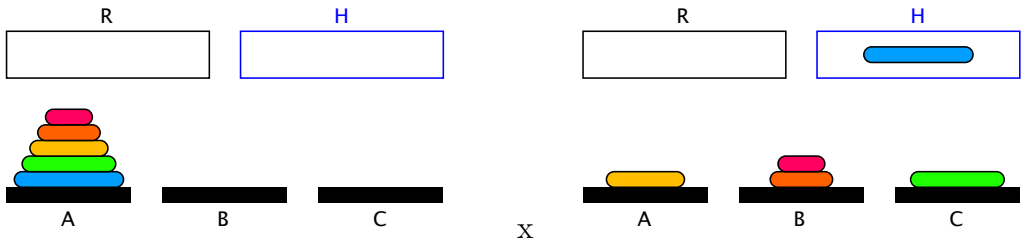
We demonstrate use of a TPN control scheme for turn-taking and the value of action interruptions. We use a particular domain, a collaboration to solve the Towers of Hanoi, as described in Section 5.2.1. Our evaluation is performed within this domain through two means. One is a traditional user study, the protocol for which is detailed in Section 5.2.3. The other is a simulation experiment made possible by the TPN, described in Section 5.2.4, which we believe is an interesting contribution of the TPN representation for HRI research.

5.2.1 Domain description

To explore human-robot collaboration while staying in the realm of our robot's cognitive, perceptual, and physical capabilities, we chose to start with the classical artificial intelligence problem of the Towers of Hanoi. We intended for this abstract toy problem to be a metaphor for a collaborative workplace scenario in which a robot and a human need to cooperate in order to accomplish a physical goal, perhaps in a repetitive fashion, but while sharing certain resources such as tools and space.

The problem of the Towers of Hanoi is described by a set of three pegs and an ordered set of N pieces, usually disks of increasing size. The goal is to move the entire set of pieces from one peg to another by moving only one piece at a time. A piece cannot sit on any piece of lower ordinality than itself. For achievable manipulation with the robot, we instead used a set of equally sized cups that differed in color and relied on a color sequence to represent the ordering.

We modified the problem slightly to form a dyadic embodied collaboration. Each agent in the dyad is permitted to pick up a single piece at a time. However, pieces do not teleport to pegs instantaneously, as pick and place actions require time to execute in the real world. Thus, the state space is a vector of length N in which each value represents the owner of the piece: either a peg $\in \{A, B, C\}$ or an agent $\in \{H, R\}$. In



(a) Starting configuration.

(b) Legal state in which the human is holding a piece.

Figure 20: Example state representations in collaborative Towers of Hanoi. A–C represent pegs, and H(uman) and R(obot) represent agents.

this formulation, the human and the robot essentially serve as extra pegs, each with a capacity of one piece.

The state space is a directed graph in which nodes are the states described above. Nodes are connected with edges representing actions to pick up a piece or place an owned piece on a specific peg, or verbal requests for the human to perform either of these actions. Figure 21 shows the first four levels of reachability in the state space. A plan is determined using Dijkstra's shortest path algorithm on the state space graph. The robot replans the solution as needed whenever state changes are detected.

Towers of Hanoi is used in cognitive psychology tasks and can already be difficult for humans with $N = 4$ [58]. For $N = 5$ in our modified collaborative Towers of Hanoi, the solution is difficult for a human to see intuitively towards the beginning. About halfway through, the solution becomes much clearer, and at this point most humans are able to see the action sequence required to reach the goal.

5.2.2 Timed Petri Net implementation of domain

Figure 22 shows a system visualization of the implemented TPN for the Towers of Hanoi. A token in $p_{experimental}$ demarcates whether the robot should run $t_{backoff}$ or not, which activates the behavior for backing the robot's arm away from the shared workspace. This runs only when there is also a token in $p_{conflict}$ deposited by $t_{detectIntent}$, which is the perceptual process monitoring whether the human's hand

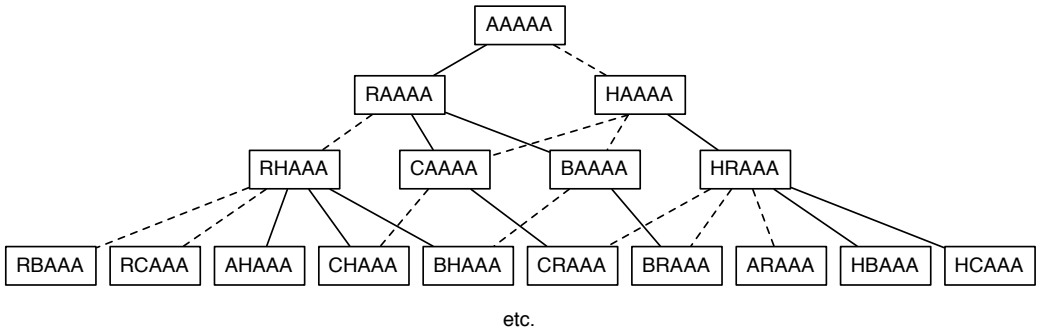


Figure 21: The first four levels of the collaborative Towers of Hanoi reachability graph for $N = 5$. Solid lines indicate manipulation actions, and dashed lines represent requests for human actions.

has entered the shared workspace.

The transitions t_{act} and t_{move} differ in that t_{move} describes physical movement, and t_{act} describes the cognitive actions of processing the task state and selecting a task action to progress towards the goal. The task action can be either a manipulation action (controlled by t_{move}) or a verbal request to the human. The place $p_{bottleneck}$ gets filled when the task state reaches a point when the robot is bottlenecking on the human performing a task action, at which point $t_{handleBottleneck}$ initiates the control chain for a verbal request by interfacing with the speech process. When a human's action removes the bottleneck, t_{thank} is run to thank the user using speech.

The places p_{moving} and p_{still} for robot motion and $p_{speaking}$ and $p_{!speaking}$ for robot speech are meta-indicators for robot state. Incoming transitions that fill these places can be thought of as listeners. That is, $t_{detectMotion}$ does not control the robot to change the state of the external world; it simply internally monitors the robot's joints to determine if the robot is currently moving or not. So $t_{handleBottleneck}$ synchronizes on p_{still} and $p_{bottleneck}$ before running, meaning that it waits for a certain duration after the robot is no longer moving and after a bottleneck has existed for some amount of time before generating a verbal request to the human.

We emphasize that the controller in Figure 22 is a *domain-dependent* and ad hoc design. Such a controller can be rapidly prototyped, but does not contain components

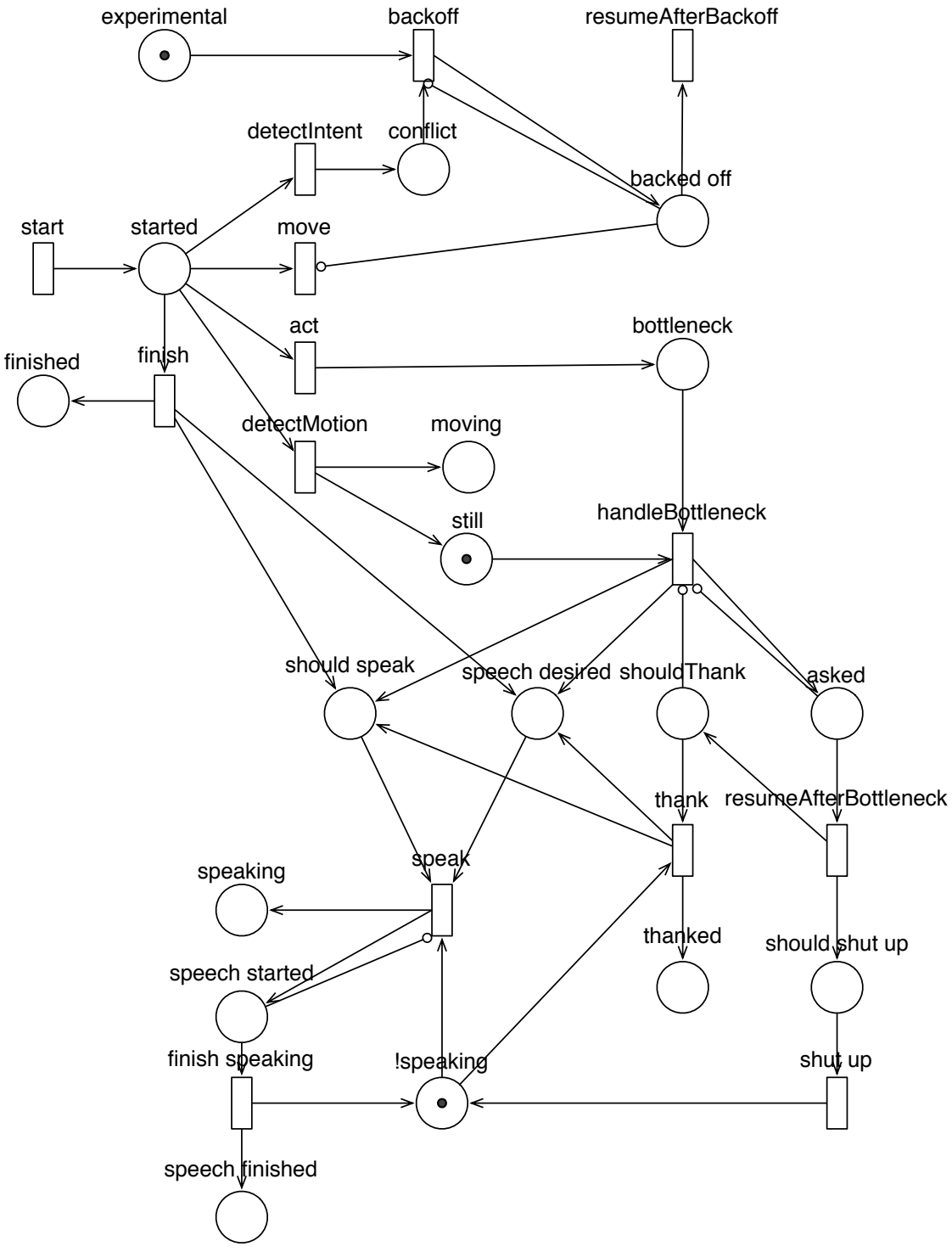


Figure 22: The system visualization of the timed Petri net used to control the robot in the Towers of Hanoi domain.

that can be clearly transferred to other situations. Later chapters are dedicated to generalizing the process of constructing such a TPN controller from modular TPN subprocesses with clearly defined interfaces.

5.2.3 User study

We designed and conducted an experiment to evaluate the effects of action interruptions within the system. The experiment was a between-groups study in which 16 participants collaborated with our humanoid robot Simon to solve the Towers of Hanoi problem.

5.2.3.1 Environment Setup

In the experimental setup, the participant stands across from the robot. The two are separated by a 34-inch square table that is covered by a black tablecloth (Figure 23). The table has three pegs rigidly affixed to a black foam board, and the pegs are equidistant from the positions of the human and the robot. The black foam board is designated as the shared workspace. The robot manipulates objects using only its right arm. Five differently colored, equally sized plastic cups are used as the Towers of Hanoi pieces. A stack of differently colored, differently sized blocks stands on a table behind Simon and serves as a mnemonic to help the participant remember the color sequence and the goal configuration.

Perception of the Hanoi pieces is done using an overhead camera pointing at the table. Because only the top piece is perceivable from this position, inferences about state need to be made based on these observations. Important perceptual events are color changes at peg locations, which could indicate any agent (the robot or the human) either removing a cup or placing a cup. Legal states consistent with the visual data and with the robot's intentions are preferred.

A structured light depth sensor, the Microsoft Kinect, is mounted on a tripod positioned adjacent to the robot and facing the human. The Kinect is registered

to the robot's world coordinate frame, and the Kinect SDK is used to detect and track the participant's head and hand positions. This allows the robot to gaze at the participant's head pose, as well as to detect where participants' hands are in relation to the pegs. Specifically, the robot detects when the participants' hands enter and leave the shared workspace, as indicated by the region of the black foam board. When a human hand is in the shared workspace, the nearest peg is assumed to be the target of the human's next manipulation action.

5.2.3.2 Experiment Conditions

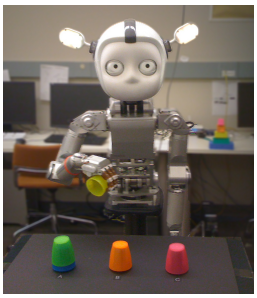
The study is a between-groups design containing two conditions. The robot operated autonomously in both conditions:

- *Interruption condition* – In this condition, the robot interrupts its actions in response to the human. When performing reaching actions towards a particular peg, if the human's hand is in the shared workspace and approaches the direction of that peg (as detected by the Kinect skeleton tracker) then the robot interrupts its reach and switches its eye gaze to the human to signal yielding intent (Figure 23(c)). When performing speaking actions to request an action from the human, it interrupts its speech if the desired state change has been detected before the robot finishes speaking.

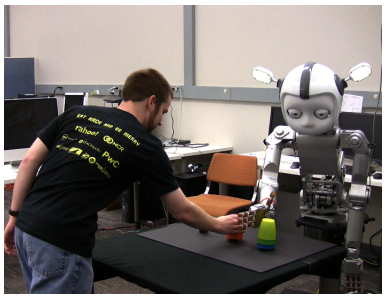
- *Baseline condition* – In this condition, the robot always runs reaching and speaking actions to completion before proceeding (Figure 23(b)).

5.2.3.3 Protocol

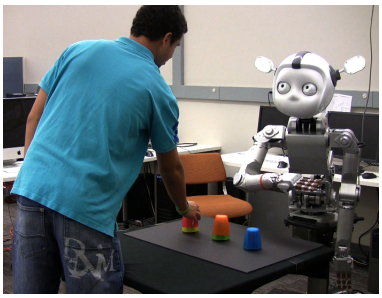
Participants in both conditions were given identical instructions. After the Towers of Hanoi task was explained, participants were told that they were going to solve it collaboratively with Simon. They were instructed to use only one arm and to move only one piece at a time. They were encouraged to try to solve and execute the



(a) Workspace from participant perspective.



(b) Baseline condition.



(c) Interruption condition.

Figure 23: Simon backs off from the shared space in the interruption condition but not in the baseline condition.

task quickly and in a parallel fashion with the robot. They were told that the robot might ask them to do some actions, but they did not have to listen to him, since the robot's world state could be prone to perceptual errors. They were also told that if Simon made any manipulation errors (e.g. failed to release a cup properly over a peg), that they should simply pick up the dropped cup and restore the state to a legal configuration.

Execution of the collaborative task lasted roughly five minutes per participant, and video was taken of the participants with their consent. Timestamped data of task start, completion, and state changes were logged throughout the interactions (and later confirmed or corrected through video analysis). After interacting with Simon, participants completed a survey containing the following questions:

1. On a scale from 1-100, how much did you contribute towards mentally solving the puzzle? (50 means both contributed equally)

2. On a scale from 1-100, how much did you contribute towards physically solving the puzzle? (50 means both contributed equally)

3. Please rate the following statements about the task. (1 = strongly disagree, 7 = strongly agree)

(a) The task would be difficult for me to complete alone.

(b) The task would be difficult for Simon to complete alone.

(c) The task was difficult for us to complete together.

(d) My performance was important for completing the task.

(e) Simon's performance was important for completing the task.

4. Please rate the following statements about the interaction with Simon. (1 = strongly disagree, 7 = strongly agree)

(a) Simon was responsive to my actions.

(b) Simon was team-oriented.

(c) I trusted Simon's decisions.

(d) I had influence on Simon's behavior.

(e) Simon had influence on my behavior.

(f) I had to spend time waiting for Simon.

(g) Simon had to spend time waiting for me.

(h) We were efficient in completing the task.

(i) The interaction pace felt natural.

(j) There were awkward moments in the interaction.

5. (Open-ended) Please provide a critical review of Simon as a team member. Imagine that Simon is being evaluated at a workplace.

5.2.4 Simulation experiment

With the same TPN system used to the control the robot in the experiment described in Section 5.2.3, we developed a simulation experiment to investigate the effects of user tendencies on the system dynamics. This experiment was conducted after a

preliminary analysis of the results from the user study, as we believe it is important for simulations to be grounded in real user behavior. Qualitative observations of participants during the study pointed to certain factors being important to the task outcome. In particular, we observed that participants strategized differently about how to approach the problem. Certain participants made frequent, seemingly exploratory actions, sometimes undoing their most recent action or holding on to a piece at a peg while scratching their chins. Others were content to stand back and wait for the robot to tell them exactly what to do. In addition, participants tended to move faster than the robot, but most were worse than the robot at planning, especially at the beginning of the interaction.

Based on these observations, we developed a simulated user as a Petri net subgraph connected to the robot control graph. The simulated user subgraph includes the following components, as depicted in Figure 27:

- $P = \{p_{acting}, p_{idle}, p_{comply}\}$, where p_{acting} indicates a move has started, p_{idle} indicates that a move has finished, and p_{comply} indicates that a robot's request should be complied with. When p_{acting} has tokens, the human hand is detected as being near the peg of the selected move.

- $T = \{t_{start}, t_{finish}, t_{comply}\}$, where t_{start} selects and starts a move, t_{finish} ends a move, and t_{comply} decides whether to comply with a robot's request.

- $I = \{p_{acting} \rightarrow t_{finish}, p_{comply} \rightarrow t_{start}, p_{acting} \rightarrow t_{start}, p_{Speech} \rightarrow t_{comply}\}$, where p_{Speech} is a place in the robot graph indicating that robot speech has started.

- $O = \{t_{start} \rightarrow p_{acting}, t_{finish} \rightarrow p_{idle}, t_{comply} \rightarrow p_{comply}\}$.

We varied the simulated user's behavior along the following dimensions:

- *Speed* – the amount of time taken by the user per move (1–6 seconds). This controlled δ_f for t_{finish}, the time tokens spent in p_{acting} before t_{finish} fired.

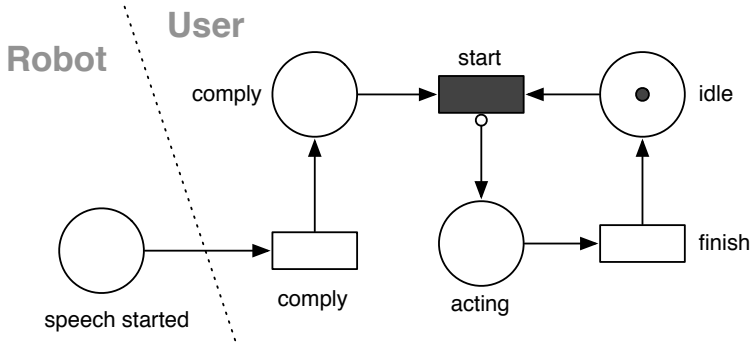


Figure 24: The simulated user behavior.

- *Initiative* – percentage of the time the user spends performing task actions in the shared space (0–50%). The transition t_{start} triggered when the following expression exceeded the initiative value, given by the intervals in p_{acting}:

$$\frac{\sum_i \tau_{\beta_i} - \tau_{\alpha i}}{\tau} \tag{1}$$

- *Compliance* – probability of the user complying with a robot's verbal request by performing the requested action (0–1), used in the control of t_{comply}.

- *Correctness* – probability of the user picking moves that are closer to the goal rather than randomly from the legal options (0.5–1), used in the control of t_{start}.

The experiment included 200 interaction runs, 100 each per experimental condition described in Section 5.2.3.2. Each run was produced by sampling parameter values uniformly at random from the specified ranges. The Petri net was run using a clock at 10x speed; that is, all actions were correspondingly sped up, including the robot motion, rate of text-to-speech, wait times, etc. The full experiment took approximately 6.4 hours to run (64 interaction hours). We terminated any given user run after 30 interaction minute, even it was not yet finished; this could occur as a result of poor strategies, timing, or deadlocking.

This experiment has two metrics of interest:

- *Execution duration* – the total time taken to complete the task.
- *Task balance* – the percentage of the final plan (action sequence) contributed by the human as opposed to the robot.

5.3 Results and discussion

The results from the user study and simulation experiment are reported here. Our hypothesis was that action interruptions would increase interaction fluency by improving the balance of control between the robot and the human. That is, by relinquishing control to the human appropriately, the robot could allow the human to make better contributions to the task. We hypothesized that such a shift in balance would improve overall task performance, which should be observed as shorter task execution times. Additionally, we expected people to have a more positive perception of the robot and of the interaction in the interruption condition.

5.3.1 User study analysis

Results from the user study indicated that action interruptions resulted in reduced task execution time, perception of increased human contribution, and perception of fewer awkward moments in the interaction.

5.3.1.1 Task Efficiency

The human-robot teams took significantly less time to complete the task in the interruption condition when participants had more control of the workspace ($M = 3.43$ minutes, $SD = 0.69$), as compared to baseline ($M = 4.56$ minutes, $SD = 1.55$), $t(7) = -1.9$, $p < .05$. Although the robot's planner attempted to minimize the number of moves, the goal was to optimize completion time, and humans were faster at executing actions but cognitively could not see many moves ahead.

There was no significant difference in plan length across the conditions, but the robot contributed significantly fewer moves in the interruption condition ($M = 8.75$,

$SD = 1.71$) compared to the baseline ($M = 13.25$, $SD = 4.47$), $t(7) = -2.81$, $p = .01$. This led to a marginally significant differences in the ratio of human moves to robot moves (interruption: $M = 1.79$, $SD = 0.87$; baseline: $M = 1.19$, $SD = 0.31$, $t(7) = -1.79$, $p = .06$). These show the human's increased control when the robot used fluent actions.

There were also significantly fewer robot verbal requests in the interruption condition ($M = 3.5$, $SD = 1.58$) than in the baseline ($M = 7.25$, $SD = 2.54$), $t(7) = -4.02$, $p < .01$. Since there was no significant difference in human compliance to requests and compliance was high overall (94%), the robot contributed even more of the solution in the baseline through these requests.

5.3.1.2 Perception of Contribution

Two survey questions concern the relative contribution of each team member to the success of the task. The relative contributions are parametrized along two dimensions: mental and physical. The mental contribution pertains to the algorithmic solution to the problem, and physical contribution pertains to the execution of it. For each dimension, we categorized people's numerical (1–100) responses about their own contributions as *equal to* ($= 50$), *more than* (> 50), or *less than* (< 50) that of the robot.

The distributions across conditions for relative physical contributions were the same; participants were well aware of their superior manipulation capabilities and movement speed compared to the robot. However, participants in the interruption condition were statistically more likely to state that their mental contribution was equal to or higher than the robot's compared to the baseline condition, $\chi^2(2, N = 8) = 6.17, p = .05$. This agrees with a result from our previous work, in which more submissive robot behavior results in better mental models for the human [20].

In the baseline condition, the robot's increased tendency to monopolize the space

above the pegs could have given the impression of always knowing what to do next. Although this was true to some extent, the robot's world state was still subject to perceptual errors, and the robot's plan did not take into account differing speeds of manipulation, so more human intervention would often have been beneficial. In addition, the yielding behavior may have communicated willingness to consider the human's strategy, making users more open to taking initiative. One participant in the baseline condition declared that the robot was *"a soloist"* and *"should be more of a team player."* Another observed, *"Simon was solving it so well... it did not feel like teamwork."* In the interruption condition, the robot's backing off from the shared space allowed the human to exert control over the plan being used to reach the goal. One participant said, *"I helped guide him to the solution to the puzzle quickly,"* asserting his belief of who was in charge.

5.3.1.3 Interaction Fluency

One of the questions asked participants to rate their agreement or disagreement, on a 7-point Likert scale, with the following statement: *"There were awkward moments in the interaction."* Although the notion of awkwardness may be difficult to quantify, we posed the question because we thought it would be intuitive for humans to answer. People in the interruption condition were less likely to agree that there were awkward moments in the interaction ($M = 3.75$, $SD = 1.71$) as compared with participants in the baseline condition ($M = 5.50$, $SD = 1.71$), $t(7) = 2.64$, $p = .03$.

We interpret this result as indicating a higher degree of interaction fluency in the interruption condition. Interruptible actions increased the impression of a reciprocal interaction, in which transparent intentions modulated behavior in both directions. A participant in the interruption condition commented: *"Simon... allowed me to make moves when I wanted while at the same time being decisive when he saw that I was pausing. He is an extremely good team member."*

5.3.2 Simulation analysis

To analyze the simulation results, we ran ANOVA for the five factors described in Section 5.2.4 to characterize the impact of and interaction between factors for the observed variables of *execution duration* and *task balance*. The experimental condition was treated as a categorical variable (presence or absence of interruptions), and the others as continuous variables. These results are reported in Table 2. There were significant main effects for the manipulation of speed, initiative, and correctness for the simulated user. This implies that purely from the system standpoint, these human factors are observably important in determining task success and relative task contribution. Compliance with the robot's requests does not appear to have a significant effect. And although one might be inclined to assume that action interruptions should automatically improve task completion time by spending less time on unnecessary actions in general, the experimental condition (using interruptible actions) alone does not seem to have a significant effect, although there is a marginally significant interaction between the presence of interrupts and the initiative of the user ($p = .05$).

These simulation results only tell the story of the dynamics of the system. What they do not indicate is the actual tendencies of real users. It still remains to be seen whether the uniformly distributed parameter space in the simulation experiment at all accurately represents a random selection of users in the real world. In addition, the essential question is whether the condition manipulation induces different behavior in users, resulting in differing distribution of such parameters across the groups.

To answer this, it is necessary to characterize the parameters of the study participants in same format as the simulation experiment. The segments of time during which participants performed manipulation actions in the shared space were annotated by two coders. The intercoder agreement was 93.5%, describing the percentage of time that the annotation matched between the coders (between manipulating or not manipulating). The sum of times spent over these segments divided by the total

Table 2: ANOVA for simulation experiment results on the factors of condition, speed, initiative, compliance, and correctness. Execution duration describes the total time taken to complete the task, and task balance describes the percentage of the final plan (action sequence) contributed by the human as opposed to the robot.

Factor	**Execution Duration**	**Task Balance**
Condition	$F(1, 199) = 0.67$, $p = .41$	$F(1, 199) = 0.65$, $p = .42$
Speed	$F(1, 199) = 23.12$, $p < .0001$	$F(1, 199) = 25.05$, $p < .0001$
Inititative	$F(1, 199) = 17.85$, $p < .0001$	$F(1, 199) = 11.15$, $p < .01$
Compliance	$F(1, 199) = 0.09$, $p = .77$	$F(1, 199) = 0.1$, $p = .76$
Correctness	$F(1, 199) = 6.36$, $p = .01$	$F(1, 199) = 11.02$, $p < .01$
Cond × Speed	$F(1, 199) = 0.96$, $p = .33$	$F(1, 199) = 0.01$, $p = .91$
Cond × Init	$F(1, 199) = 3.61$, $p = .05$	$F(1, 199) = 0.16$, $p = .69$
Cond × Comp	$F(1, 199) = 0.11$, $p = .75$	$F(1, 199) = 0.32$, $p = .57$
Cond × Corr	$F(1, 199) = 0.06$, $p = .80$	$F(1, 199) = 0.48$, $p = .49$
Speed × Init	$F(1, 199) = 12.8$, $p < .001$	$F(1, 199) = 22.98$, $p < .0001$
Speed × Comp	$F(1, 199) = 0$, $p = .96$	$F(1, 199) = 1.21$, $p = .27$
Speed × Corr	$F(1, 199) = 10.7$, $p < .001$	$F(1, 199) = 8.71$, $p < .01$
Init × Comp	$F(1, 199) = 0$, $p = .98$	$F(1, 199) = 0.08$, $p = .78$
Init × Corr	$F(1, 199) = 35.68$, $p < .0001$	$F(1, 199) = 12.99$, $p < 0.001$
Comp × Corr	$F(1, 199) = 0.04$, $p = .85$	$F(1, 199) = 0.17$, $p = .68$

task time yielded the initiative parameter; the average of this value was taken between the two coders. To determine the average time taken per action (speed), the number of actions per segment was also annotated with discussion between coders to resolve differences. Correctness was determined by whether the resulting game state of each human action was closer to the goal than the preceding state; the sum of closer actions was divided by the number of human actions. Compliance was determined by whether the next action taken by the human after a robot verbal request was the requested action, divided by the number of requests.

The resulting parameter values are shown in Table 3, and a comparison of the parameters tested in simulation and the parameters of the human participants is visualized in Figure 25. As can be seen from the figure, the parameters of participants from the user study were well within the span of values tested in simulation. As shown in Table 3, speed, compliance, and correctness did not differ across the groups,

but initiative differed significantly. On average, subjects in the interruption condition spent 50% more of the total task time performing actions as compared to the baseline condition. We thus conclude that the presence of action interruptions leads users to take more initiative in the task, leading to the observed increase in task efficiency and improved balance of control.

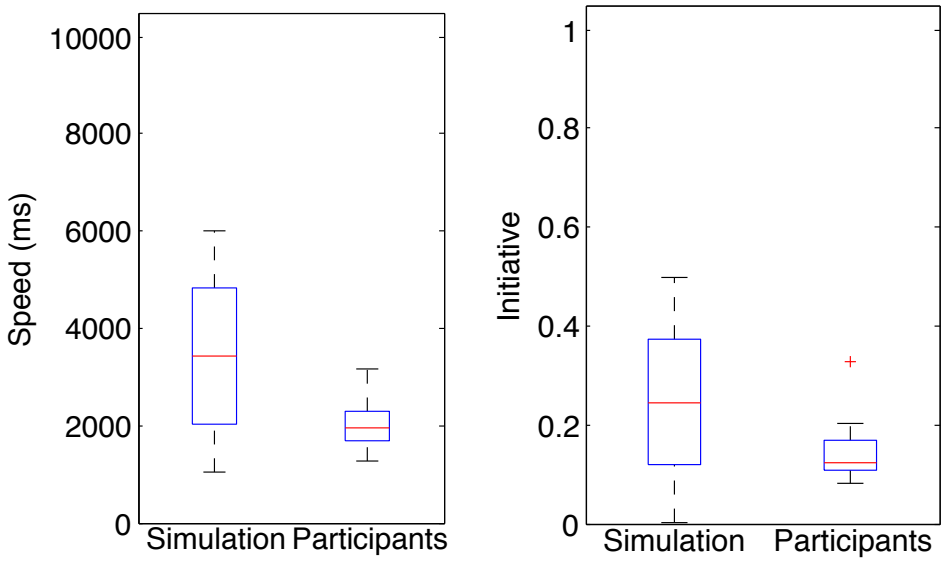
Table 3: Shown are the speed, initiative, compliance, and correctness for the participants in the user study across the two conditions. The parameter values were determined from logs and video coding

Parameter	Baseline	Interruption	Significance
Speed (secs)	1.6 (0.2)	2.0 (0.4)	$t(7) = -1.32$, $p = .11$
Initiative (%)	10.8 (2.2)	16.2 (6.9)	$t(7) = -1.96$, $p < .05$*
Compliance (%)	96.9 (8.3)	90.1 (21.8)	$t(7) = -0.79$, $p = .23$
Correctness (%)	71.1 (16.1)	79.5 (15.6)	$t(7) = -0.50$, $p = .32$

5.3.3 Generalizability of results

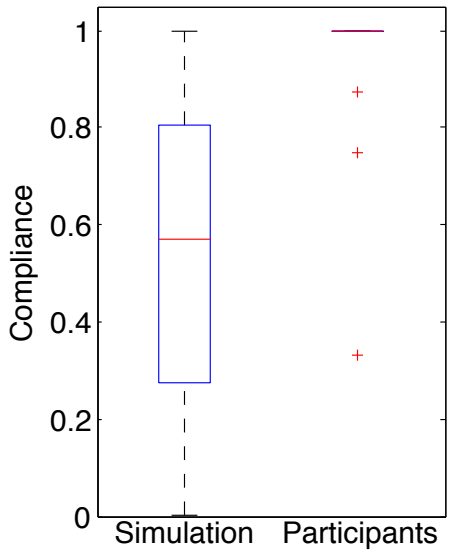
In this domain, we have demonstrated an example of how our system simulation can augment the analysis of a user study. An obvious caveat with all simulation work is that assumptions made in simulation may not generalize well to interactions with human participants in the real world. All of human behavior cannot be summarized in a handful of parameters, and the parametrizations can be oversimplifying. There is also always the question of whether it is worth channeling effort into developing more accurate user models when user studies are needed anyway.

However, we do believe that iterative analysis of user data and simulation data can provide more perspective into how certain results were attained. The ability to run large quantities of simulations in much less time than would be needed for user studies is a powerful tool for developing and understanding the robot system. It also allows for broader coverage of parameters that may occur less commonly in the recruited user populations, which can be useful for stress-testing or finding corner cases. Because human-robot interactions are so complex in state and timing, they

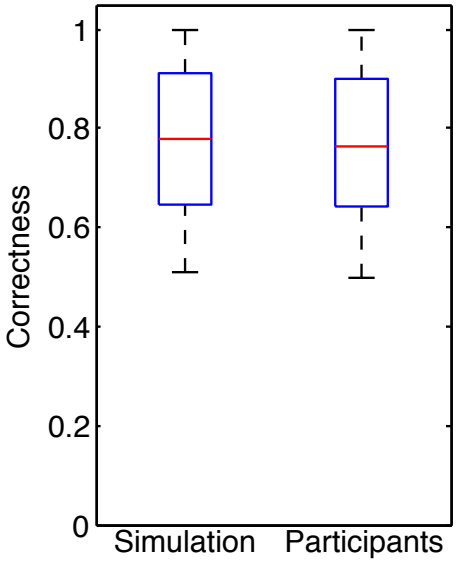


(a) Speed

(b) Initiative



(c) Compliance



(d) Correctness

Figure 25: Comparison of user parameters between those sampled as inputs in the simulation experiment and those exhibited by human participants in the user study.

tend to be difficult to pigeonhole into closed-form mathematical techniques. We think that system characterizations through simulation can offer a scalable way to understand fluency in the face of such complexity. In this work, we have shown the ease with which a TPN framework allows such simulation experiments.

Another issue concerns the generalizability of our specific domain to other collaboration scenarios. One could argue that the inability of the robot to optimize for time rather than plan length was a failure of foresight and programming. In addition, the robot was slower and less dexterous than the human, which may not generalize to tasks where the opposite is true; in those cases, could the human's increased control be detrimental? It certainly should not be assumed that our outcome of reduced execution time universally generalizes to future tasks. However, the simulation technique allows for the rapid analysis of any adjusted dynamics. For example, if it became possible for the robot to move faster than the human, then the experiment could be quickly rerun after modifying the controller. In addition, any preprogrammed optimization ability of the robot may not optimize the human's goals at a particular moment. Humans adapt easily to fluctuating goals, and allowing robots to be able to yield appropriately results in spontaneous flexibility of the dyadic system, providing users greater control over this adaptive optimization process.

5.4 Summary

Towards the goal of developing robot behavior that improves fluency in multimodal reciprocal interactions, we describe the design and implementation of a system for the control and analysis of timing in turn-taking interactions based on a timed Petri net representation. The system focuses on the skill of yielding resources to the human and is demonstrated autonomously in a human-robot collaboration scenario based on the Towers of Hanoi. To examine the role of interruptions, we employ a novel evaluation mechanism combining two types of experiments. We ran a user study

with 16 participants in which the robot was autonomously controlled by our system, and we ran a simulation experiment that simulated 200 such users parametrized by the factors of speed, initiative, compliance, and correctness. Our analysis of results from both experiments showed that our implemented action interruptions increased task efficiency primarily through a mechanism of increasing the initiative of the human partner. This resulted in the perception of improved interaction balance, which subsequently led to a reduction in time needed to complete the task.

To complement this implementation and characterization of *yielding* resources, we next turn to the problem of when it is appropriate for a robot to *seize* resources.

CHAPTER VI

FLOOR REGULATION

The turn-taking process can be construed as comprising four components of floor regulation: *seizing* the floor, *yielding* the floor, *holding* the floor, and *auditing* the owner of the floor. The previous chapter argues strongly for robots able to *yield* control to humans, but still a central problem remains: the timing of when to *seize* the floor. A robot that only yields does not take initiative to recover from lapses, introduce new information or action, or provide backchannel feedback. The challenge, then, is how to reintroduce initiative into a robot's turn-taking without inappropriately detracting from the human's control of the interaction. From a usability perspective, the system should be held responsible for taking initiative to structure the interaction in order to recover from moments of ambiguity or confusion, as well as to make the interaction state self-evident. In addition, different scenarios may require different levels of initiative from the human and the robot, necessitating an interaction architecture that accounts for such variable control.

Prior work in human social psychology has shown how dominant or deferent conversational styles correlate with influence over a task [71]. We thus posit that one's implementation of these turn-taking behaviors for a robot significantly affects the overall social dynamics of the dyad. Turn-taking is characterized by a fundamental tension between who is initiating action or communication (seizing and holding resources), and who is being supportive and contingent upon the other (yielding resources and auditing the interaction partner). For humans, the decision of when to seize and how often to seize often differs depending on the relative status of the interaction partners. Nearly all social transactions between humans are defined by relative

status, and communicating the appropriate status can be essential to the success of the interaction [52].

This chapter describes a parametrization of turn-taking that enables the robot to control its behavior for achieving a desired social dynamic. In Sections 6.2 and 6.3, we present an experiment within a highly open-ended domain of a robot and a human playing together with toys on a tabletop. Our results in Section 6.4 highlight some of the effects that exhibiting different personality or status behavior can have on humans. The experiment is designed to explore the range of interaction dynamics made possible by CADENCE, and thus features two somewhat extreme parametrizations rather than attempting to test any hypothetical "optimal" setting. Realistically, the most suitable parametrization will vary significantly for different social contexts (a discussion point we raise in Section 6.5.3). We emphasize that the experiment and results do not aim to prescribe or advocate any specific social dynamic for HRI, but simply to characterize interaction differences between contrasting turn-taking styles made possible by the system.

We start by describing implementation of parametrized floor regulation as a generic system in Section 6.1. This is followed by a description in Section 6.2 of how we instantiate this model for turn-taking interactions in the particular domain of playing with objects on a tabletop. We use this domain to evaluate the system in an experiment with 30 human participants. Our results show that: (1) manipulating our floor regulation parameters results in different robot behavior; (2) people perceive this difference in behavior and attribute different personalities to the robot; and (3) changing the robot's personality results in different behavior from the human, manipulating social dynamics of the dyad. Section 6.5 then identifies system shortcomings and applications to be addressed in future work.

6.1 Turn-taking process

The TPN controller in Chapter 5 was able to interrupt its actions, but it was constructed in a fairly task-dependent way. Here we seek to develop a more general-purpose floor model. The conversational floor, the focal point of turn-taking in linguistics, is a shared resource due to the cognitive difficulties of processing simultaneous speech [6]. In addition, the floor owner has the opportunity to exhibit greater control over additional shared resources such as shared physical space and objects through accompanying gesticulation and manipulation actions. Floor exchange represents shifting control over the outcome of an interaction.

In this section, we describe the components of the turn-taking Petri net that are used to monitor the floor and regulate its ownership. Because the Petri net behavior model is a flat model comprising many connected subgraphs, we have annotated common nodes between Figures 26, 27, and 28 with a consistent shading scheme in order to highlight the connections between the subgraphs.

6.1.1 Floor state representation

The cornerstones of turn-taking behavior are seizing the floor, holding the floor, yielding the floor, and auditing the current owner of the floor. CADENCE models seizing and yielding as transitions (t_{seize} and t_{yield}) that lead to the states of holding ($p_{holding}$) and auditing (represented by $p_{yielded}$, which activates t_{audit}). This execution flow is discussed more in Section 6.1.3. The balance between the time one spends exhibiting holding versus auditing behavior is critical to the social dynamics of a turn-taking interaction. The combinations of the user's and robot's attempts at holding and auditing additionally result in the meta-states of *conflict* (both taking a turn), a *lapse* (neither taking a turn), or one or the other owning the floor.

Figure 26 depicts the relationship between engagement, individual turn states, and dyadic floor states in the system. Each boxed area indicates a set of places

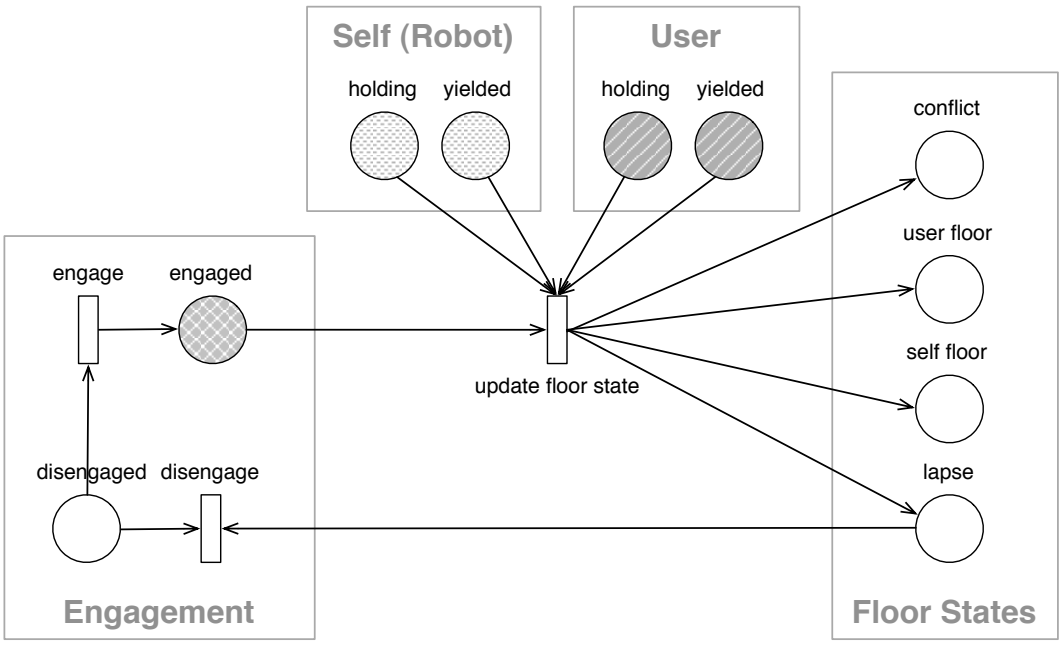


Figure 26: This diagram shows the relationship between engagement, turn states for the robot and the user, and dyadic floor states. The floor state update is based only on the time that $p_{holding}$ and $p_{yielded}$ from the *Robot* and *User* processes contain tokens. The full *User* model is shown in Figure 27, and the full *Robot* turn-taking control process is shown in Figure 28.

between which a single token is shared; hence, these places are mutually exclusive. The dyadic floor state is only updated if the robot is currently engaged with an interaction partner. This floor state is determined as a function of the time that the robot and the user have spent in their respective current turn states. For example, simultaneous holding that exceeds a duration referred to as *conflict time* results in a dyadic state of conflict, and similarly for *lapse time* and lapses. The dyadic floor states then drive other decisions made in the robot's turn-taking, such as whether to interrupt itself or take more initiative. Parameters related to conflicts and lapses are summarized in Section 6.1.5.1.

6.1.2 User modeling

The perceptual signals used to monitor the user's behavior include speech presence, gesturing, and whether the user is gazing at or away from the robot. The particular implementation of these signals can be expected to vary across domains. Details for one relatively domain-invariant implementation of these signals are given in Section 6.2.3. These low-level features are fundamental to attaining and communicating attention in the visual and auditory channels.

Figure 27 shows the process that models the user's turn-taking state. While the user is engaged, the firing function of t_{signal} accesses the belief system to classify perception of the user's speech and motion into the three states of $p_{suppressing}$, p_{idle}, and $p_{signaling}$. These three places share a mutually exclusive token assigned by t_{signal} at each clock cycle that indicates whether the user's actions appear to be suppressing robot turn attempts ($p_{suppressing}$), the user is not performing actions (p_{idle}), or that user action has been perceived at a lower strength or consistency than would be interpreted as suppression ($p_{signaling}$). For efficiency in this pattern of TPN substructure, a single token can be moved between the three mutually exclusive places (rather than spawning and destroying tokens at each state change). The place $p_{signaling}$ owns the token as a precursor to $p_{suppressing}$ owning it; this is used in the robot's control process to determine whether the robot should hesitate while taking a turn (see Section 6.1.4).

The places $p_{holding}$ and $p_{yielded}$ in the user model also share a token, a structure that is mirrored in the robot's control process (see Figures 26 and 28). While the place $p_{holding}$ has a token, the transition $t_{segment}$ determines whether the user is currently inside or outside of a speaking segment. This allows the modeling of states in which a user appears to be holding the floor through gesture or gaze cues but is not currently speaking (for example, after a statement of "Um...", which is typically followed by a pause and a longer spoken turn) [31].

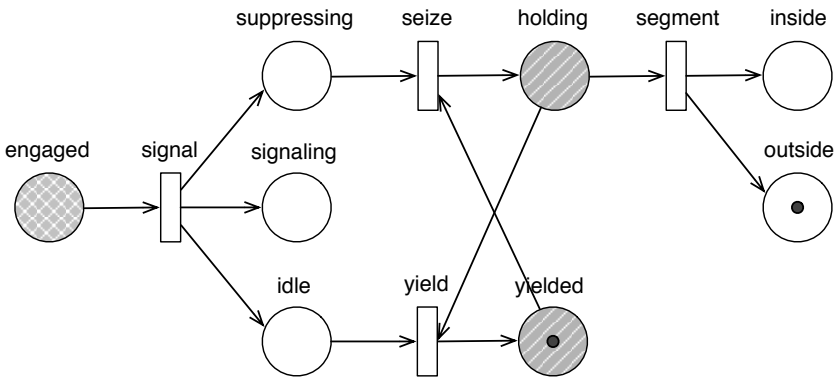


Figure 27: The user state model is based on perceptual signals for the user speaking, gesturing, and gazing away or at the robot. The places $p_{holding}$ and $p_{yielded}$ are used in conjunction with those of the robot to determine the dyadic floor state, as shown in Figure 26.

6.1.3 Full turns versus backchannels

CADENCE makes a distinction between turn-taking style and turn-taking content. The turn-taking control mechanisms for seizing, holding, yielding, and auditing as described in Sections 6.1.4 and 6.1.5 function to regulate the flow of turn content. The turn content itself falls into the categories of either full turns or backchannels in the system. Execution chains for both are depicted in Figure 28. The interruption chain for full turns is addressed in more detail in Section 6.1.4.

In each of these execution chains in Figure 28, the tokens and places are of type `Turn`. This construct is defined to comprise a set of acts (of type `Act`), where an act is a modality-specific action that can be started and stopped. Each act is associated with a function that returns the act's start time, an offset defined relative to the beginning of the turn. Thus, while the robot is holding the floor by running the transition t_{hold}, the turn-taking process delegates each act to its modality-specific execution process, each of which is a connected TPN subgraph that handles act execution and interruption correctly for the resources that it controls [26]. Running t_{yield} causes modality-specific execution to interrupt current acts and abandon future acts in the Turn. CADENCE act types currently include `SpeechAct`, `GazeAct`, `GestureAct`, and

`ObjectAct` for manipulation (some examples are given later in Section 6.2.2). Since the floor regulation model is intended to be generic, it relies on a separate context model to provide context-appropriate turn content. Sometimes default turn-taking behavior is specified, which can be overridden by acts; for example, if no `GazeAct` is specified for a given turn, the default behavior of the turn-taking process is to gaze away from the partner's face while taking a turn and to gaze back at the partner's face when yielding the floor.

Backchannels are a special form of action that communicate a speaker's current desire for, or avoidance of, the floor. Originally, backchannels were considered to be behavior that supported and maintained engagement with the current floor owner [128]. However, subsequent analyses of backchannels have showed a diversity of purposes [7, 53]. There can also be ambiguity in the difference between backchannels and short-duration full turns. For example, the same spoken utterance of "uh huh" can denote a semantic affirmative or simply an acknowledgment that the other speaker said something, depending on the particular context and speakers.

CADENCE currently supports backchannels as either continuers or incipient speakership markers, which communicate contrasting intentions towards floor ownership. The *continuer* is used when auditing to communicate that the current floor owner should continue holding the floor. A commonly used continuer in English is "mhm." The *incipient speakership marker* is used to acknowledge that the floor owner has held the floor for some time and to communicate a desire or intention to seize the floor. An example occurring in English is the construct, "Yeah, but..." followed by a full turn.

In the system, the backchannel is a subclass of Turn because it comprises the same types of acts, such as head gestures and spoken utterances. However, backchannels are run through a separate control chain in the turn-taking process. This design decision was made so that time spent backchanneling would not count against floor time.

Another reason is that these backchannels are not interruptible and do not communicate domain-specific information, which contrasts with the full-turn execution chain. After performing our system evaluation, we consider that some of these assumptions may need to be revisited; these points will be raised in Section 6.5.1.

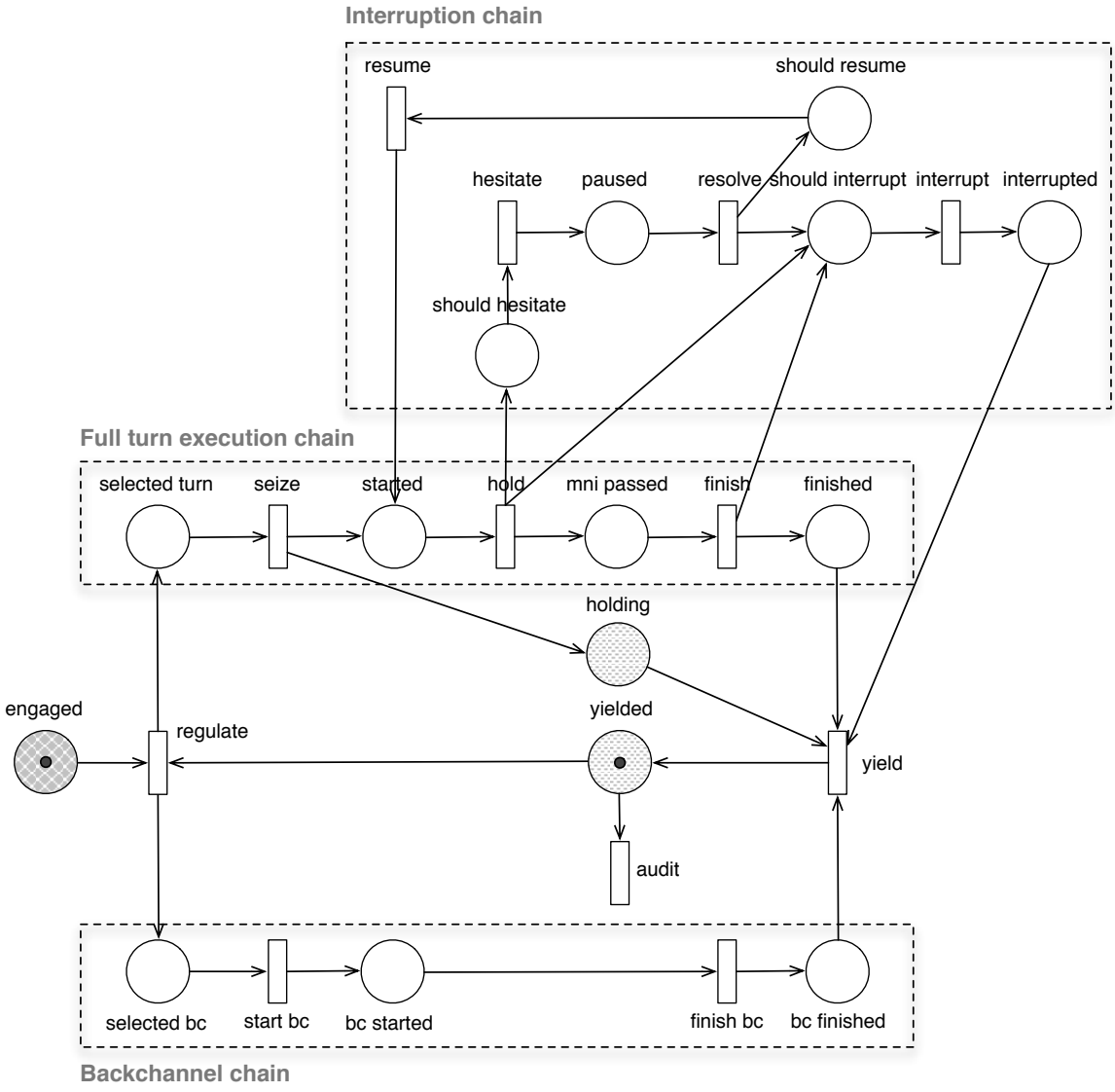


Figure 28: This diagram shows control chains for the robot's turn execution. The primary control chain is the full-turn execution chain, which is used for the playback of full turns. A full turn is moved through the interruption chain if the robot determines at some point while holding that it needs to yield the floor. The backchannel chain is an abbreviated alternative control flow for short, uninterruptible turns that do not convey domain information and may overlap more freely with human speech.

6.1.4 Yielding and auditing

In previous work, we investigated the role of a robot's self-interruptions as a mechanism for yielding the floor [26]. Robot self-interruptions are possible in our system when the *interrupt self* parameter is set to true.

We extend our previous work on action interruptions by including hesitations as a precursor to the robot's interrupting its turn. This logic is shown in the interruption chain at the top of Figure 28. The motivation for this addition was the observation that completely aborting the current turn was too extreme of a reaction when responding to short signals from the human. For example, sensor noise or fidgeting from the human could cause an extended manipulation action to abort. To reduce this level of commitment, when t_{hold} is active (indicating that the robot is taking a turn), the robot also decides whether or not to hesitate. This decision is based on $p_{signaling}$ from the user process owning a token, as described in Section 6.1.2 and shown in Figure 27. Hesitating results in pausing the current turn, which pauses active acts and prevents later acts in the turn from starting. Within a small *hesitation resolution* deadline, the robot must then decide whether to interrupt or resume its turn. If the user state transitions to $p_{suppressing}$ before the deadline, the robot proceeds to interrupt itself; otherwise, it resumes the turn. In practice, the change in the robot's behavior resulting from hesitation provides feedback to the user that some signaling was detected and allows the user to decide whether to back off or try seizing the floor.

The pausing and resumption behavior for acts varies depending on a particular act's modality. For gesture and manipulation, the robot pauses by maintaining its current pose. For gaze acts, the robot looks at the human's head when pausing and returns to the gaze act target when resuming. For speech acts, the robot stops speaking just as it would when fully interrupting itself. However, if a speech act is resumed, the robot starts again at the beginning of that particular speech act. This

works well when turns comprise multiple speech acts in the form of phrases or turn construction units [94], which we implement for our contextual instantiation described in Section 6.2. In the linguistics literature, the retake of these utterances is sometimes referred to as "recycled turn beginnings" [97].

CADENCE also supports mechanisms for turn interruption from our prior work. When running t_{hold}, the robot may skip directly to interrupting its current turn if the dyad has been in a state of conflict for some amount of time, which we define as the *conflict tolerance* parameter. The robot may also interrupt its turn after the point of *minimum necessary information (MNI)* has passed and the user is ready to proceed, which indicates that the goal of the turn has been achieved [25, 110]. This shaves time off of the ends of turns in order to support increased fluency when the dyad is well-practiced.

After the robot has yielded the floor, whether through a mechanism of interruption or from completion of turns or backchannels, the robot runs the transition t_{audit} to control behavior that supports the user's holding of the floor. This involves gazing in directions appropriate for establishing joint attention within the context, such as the user's hands, as well as periodic glances at the floor holder's face. The approach taken here for auditing agrees with the model proposed by [48], which was derived from video analysis of human-to-human turn-taking interactions.

6.1.5 Seizing the floor

After having yielded the floor, the robot must decide when to take another turn. Its moment-to-moment options are to take a full turn, to backchannel, or to wait another cycle and delay the decision. In Figure 28, the transition $t_{regulate}$ is responsible for making this decision and placing the result in $p_{turn-selected}$ or $p_{selected-bc}$ if a turn or backchannel is selected. In making this decision, the robot tries to maintain a *floor factor* k_f, which describes the ratio between the robot's holding of the floor and the

user's holding of the floor. That is, given all intervals $[r_\alpha, r_\beta)$ denoting $p_{holding}$ owning a token for the robot and $[u_\alpha, u_\beta)$ denoting the same for the user, we define a floor factor difference on a turn T with duration L_T:

$$\Delta(T) = \left| k_f - \frac{L_T + \sum_i r_{\beta_i} - r_{\alpha i}}{\sum_j u_{\beta_j} - u_{\alpha j}} \right| \tag{2}$$

Similarly for a backchannel BC, we define the difference as:

$$\Delta(BC) = \left| k_f - \frac{\sum_i r_{\beta_i} - r_{\alpha i}}{L_{BC} + \sum_j u_{\beta_j} - u_{\alpha j}} \right| \tag{3}$$

These values, in addition to $\Delta(W)$ using Equation 3 for the option of waiting to delay the decision until the next cycle, are used in the regulatory decision-making process. The relevance of a full turn, backchannel, or delay at any given moment additionally depends on multiple conditions with timing constraints. The details of this process are depicted in Figure 29. Paths throughout the tree in the figure are terminated with the following strategies for seizing or avoiding the floor, resulting in selection of backchannels or full turns:

- **Response** – a full turn that is taken in response to the user's previous turn. This occurs after a duration known as the *response delay* has passed since the user yielded the floor.

- **Deflection** – a backchannel continuer that is taken in lieu of a response, in order to avoid seizing the floor. Backchannels cannot occur more often than a period defined by the *backchannel spacing* parameter.

- **Support** – a backchannel continuer inserted between the user's speaking segments while the user continues to hold the floor in other modalities. This conveys support for the user's floor ownership.

- **Deep interruption** – a full turn that is started while the user is holding the floor. The parameter *interrupt user* must be set to true for this strategy to be used, and the user must have been holding the floor for at least a duration referred to as *interrupt patience*.

- **Interjection** – an incipient speakership marker followed by a full turn. Like deep interruption, *interrupt user* must be set to true, but interjection occurs between user speaking segments (such as the support strategy) rather than during them.

- **Lapse recovery** – a full turn that is taken after having been in a lapse for longer than a duration referred to as the robot's *lapse tolerance*. This allows the robot to take initiative to recover from a period of awkward extended inaction. Lapses longer than 3 to 4 seconds are associated with lower communicative competence in humans [124].

For all strategies based on full turns, the setting of the *require gaze to seize* parameter to true further restricts the robot's seizing of the floor to moments when the user is gazing at the robot. This is motivated by related work on the role of gaze in releasing the floor to the next speaker [35, 81]. If none of these conditions is satisfied, the robot waits without selecting a full turn or a backchannel, and the decision is repeated again at the next clock cycle.

6.1.5.1 Turn-taking parameters

As a summary, the following set of system parameters controls the dynamics of the turn-taking system, which results in different turn-taking styles. The majority of these are values or ranges of time specified in milliseconds. The purpose of each parameter is defined below with an explanation for its expected impact on interactions. In our experimental evaluation described in Section 6.3.1, we specify settings of these

parameters for producing two contrasting robot behavior styles. The parameters that differ across the conditions in this experiment are demarcated below with an asterisk.

- *Floor factor** – the robot's desired ratio of itself holding the floor to the user holding the floor.
- *Response delay* – how much time to wait after the user yields the floor before seizing the floor. Rather than being necessary for computational resource reasons, this value is used to determine how much opportunity to allow the user to seize the floor again after yielding.
- *Interrupt user** – whether the robot can try to seize the floor when the user is holding the floor.
- *Interrupt self** – whether the robot hesitates or interrupts its current turn if

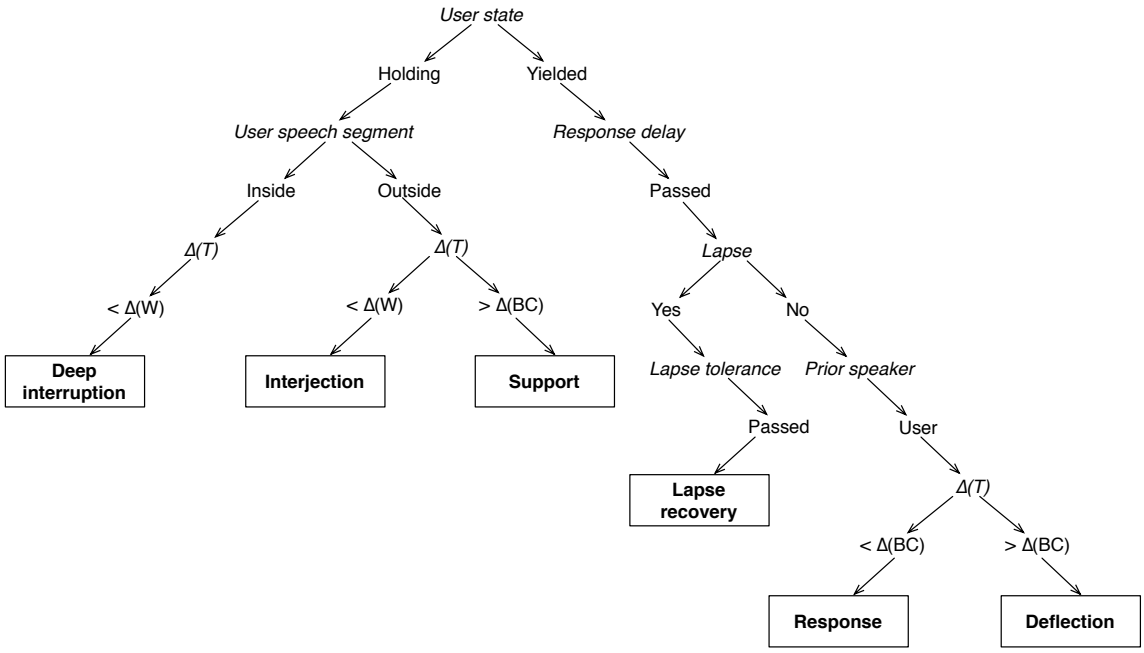


Figure 29: The decision-making process inside of the transition $t_{regulate}$ regulates floor ownership based on differences from a predetermined *floor factor* parameter that relates the robot's and the user's holding of the floor. Other conditions additionally constrain the selection of a full turn or backchannel, which leads to placing a token in $p_{turn-selected}$ versus $p_{selected-bc}$.

the user tries to seize the floor.

- *Conflict time* – how much time both the robot and the user spend continuously holding the floor before the robot considers the dyad to be in a state of conflict.
- *Lapse time* – how much time both the robot and the user spend continuously auditing before the robot considers the dyad to be in a lapse.
- *Conflict tolerance** – how much time the robot tolerates a state of conflict before being forced to interrupt its current turn.
- *Lapse tolerance** – how much time the robot tolerates being in a lapse before being forced to seize the floor.
- *Interrupt patience* – the minimum amount of time the user has spent continuously holding the floor before a deep interrupt is allowable. This parameter is used only if interruptions of the user are permitted.
- *Hesitation resolution* – the deadline after the robot hesitates that the robot must decide whether to resume or interrupt its currently paused turn.
- *Act spacing** – a uniformly sampled range of time that separates acts within a turn. Higher values encourage interjections from the user.
- *Backchannel spacing** – a uniformly sampled range of time that separates consecutive backchannels.
- *Require gaze to seize** – whether the user must be gazing at the robot in order for the robot to seize the floor.

6.2 Contextual Instantiation

A central challenge in the design of an integrated turn-taking system is the role played by context. A system that regulates turn-taking inseparably from domain specifics

has little utility, as it must be redesigned for each new domain. Thus, we strive in the design of our turn-taking controller to minimize the effort required for this transfer. In our system, the aim is for the robot's behavioral processes to be reusable across domains, and the context model is a TPN subgraph connected to these behavioral processes that gets replaced for each new domain [26].

Related to the notion of abstracting the skill of turn-taking from domain-specific knowledge is the sensitive balance between using bottom-up and top-down perception for driving robot behavior. As an example showing this contrast, a top-down perceptual process might be a grammar-based speech recognizer that relies on endpointing before returning results, while its bottom-up counterpart may be a filter for the presence of speech in the audio signal. Of course, the spectrum of semantic knowledge encapsulated by differing perceptual techniques is more fluid than these extremes.

To characterize the floor regulation system, we deliberately design an experimental context to be as open-ended as possible to focus on turn-taking driven by bottom-up signals. This includes a decision to have the robot speak an artificial language to circumvent top-down speech recognition. Perhaps contrary to intuition, this design actually allows us explore a space of more complex interactions, as compared to domains bounded by task constraints that we have used in previous work. We find that this open-ended domain uncovers the innate sense of obligation to speak, act, or yield that is driven by a human's intuition for turn-taking without being complicated by issues in task and natural language understanding. Later in this thesis, we address some of the ways that turn-taking behavior depends on semantic understanding and dialogue acts.

In this section, we describe the implementation of contextual components for this domain, to be used in an experiment with users described in Section 6.3.

6.2.1 Setting

The interaction setting is intended to support a relatively open-ended multimodal dialogue about toys at a tabletop. Participants have access to a bin of objects containing toys such as blocks and small plush animals, which they use to play with the robot. The catch is that the robot and the human do not speak the same language, so the domain is free from task semantics or the need for natural language understanding of the user's turns.

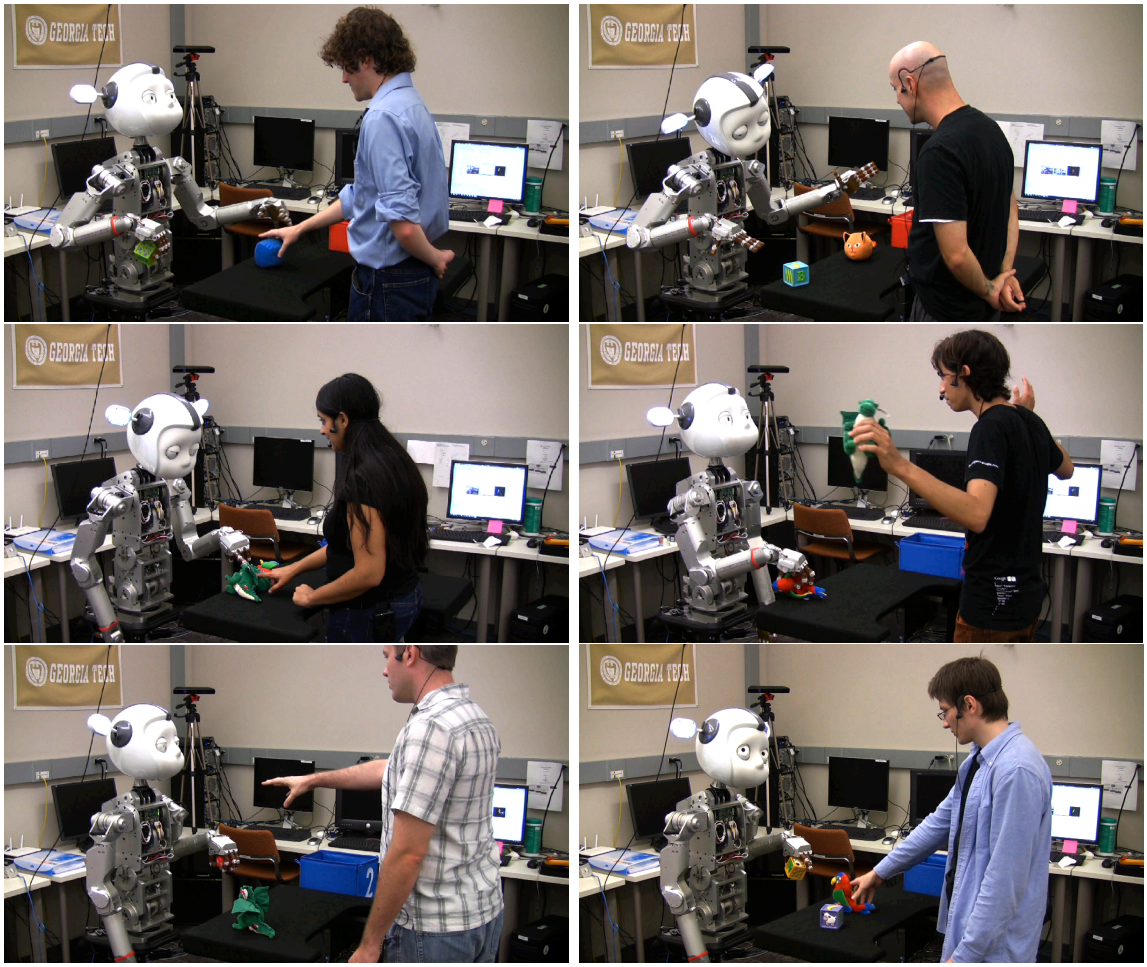


Figure 30: Examples of participants interacting with Simon in the context of tabletop object play.

6.2.2 Robot actions

Robot turns in this context are constructed as random combinations of acts in the modalities of speech, gesture, manipulation, and gaze. Figure 30 has examples of the resulting behavior. Full turns contain the following types of acts:

- A `SpeechAct` in this domain comprises phrases in an artificial language. These phrases were pre-generated by sampling random strings of phonemes. The phrases vary from approximately 1–5 seconds in duration and are grouped by the prosodic endings of ellipsis, exclamation, interrogation, and statement. Each turn consists of 1–3 of these phrases, of which the last phrase is always one of either exclamation, interrogation, or statement, and its antecedents all have elliptical prosody. The *act spacing* parameter is used to set the timing for these phrases.

- `GestureAct`s include head gestures and arm gestures. Head gestures include a head nod (looks like "yes"), a side-to-side head shake (looks like "no"), and several for communicating uncertainty through head tilt and sideways eye motions.

- Arm gestures are animations previously retargeted from human motion capture that were selected based on their interpretability as attitude towards an object or event. The communicative intentions of the human performing the gestures were shrugging, "aww shucks," "phooey," and presentation.

- An object-directed arm action (`ObjectAct`) is used to pick, place, or point at objects on the table. A manipulation action is accompanied with gaze toward the object of reference unless another `GazeAct` is specified for the turn.

- A `GazeAct` toward one of the objects on the table may also be selected; thus, an arm or head gesture can be interpreted as being directed toward the object.

Backchannels were restricted to head nodding or shaking gestures and 1–3 phonemes sampled from a limited phoneme set. Incipient speakership markers were sampled from the space of English vowels. Continuers were sampled from the consonants /m/, /n/, /h/ and the vowel /ʌ/. These different phoneme sets were intended to show the backchannels' contrasting functions, reflecting the different backchannel distributions that occur in natural languages [53, 28].

6.2.3 Perception

A Microsoft Kinect was used for tracking the human's skeleton using the Kinect Software Development Kit. Specifically, the human head and both hand positions were used for the robot's auditing behavior. The head and shoulder positions relative to the participant's hips were also used to determine whether participant gaze was oriented toward or away from the robot's head. The human was considered to be gesturing if either of the hands were in motion over the past 800 milliseconds, or if the hands were outstretched over the table. These cues were used in t_{signal} in Figure 27 of the user model.

The signal for the presence of user speech was detected through a Pure Data module [87] for determining the pitch of an audio signal. This signal was used in t_{signal} and $t_{segment}$ in Figure 27. Participants wore a headset with a directional microphone to minimize the detection of the robot's speech. The audio signal was preprocessed with an amplitude filter that was tuned to ignore the robot's voice.

To detect tabletop objects for attention and manipulation, tabletop segmentation was performed using an overhead Asus Xtion. This object perception was domain-specific and thus occurred within the swappable context model process. The table was detected using a plane extraction technique and subtracted to yield 3D point clusters representing the objects [115]. Only clusters detected within the boundaries of the table plane and above the table were considered in the context. Additional

tracking was performed to reason about occluded objects from the perspective of the Asus sensor using knowledge of both robot kinematics and human kinematics (from the Kinect skeleton). Clusters within a certain distance from robot or human link positions were not considered to be objects, but previously detected clusters exceeding a distance from agent links were considered to be occluded by agents and thus preserved.

6.3 Experiment: Open-ended play

To evaluate the preliminary implementation of CADENCE as described in Section 6.1, we designed a between-groups user study in which our robot Simon used the system to control its autonomous behavior within a situated dialogue about objects. The primary purpose of this experiment is system evaluation. To validate that our parametrized model of floor regulation is effective in achieving different social dynamics, we compared the behavior exhibited by the robot across highly contrasting parameter settings, as well as the effects of these differences on user behavior and user perceptions. The experiment also enables us to analyze any turn-taking errors that occur in either of these parametrizations, for insights on future work.

6.3.1 Parameter groups

The user study contained two conditions designed to investigate situations in which a robot shows different levels of initiative or control:

- *Active condition* – The robot tries to act twice as often as the human and deliberately interrupts the human to maintain this ratio.
- *Passive condition* – The robot tries to act half as often as the human, hesitates and interrupts its own actions, and often backchannels to avoid seizing the floor.

Table 4 shows the specific parameter differences between the conditions. We recognize that these settings represent only two points in the large space of turn-taking styles possible.

Table 4: Parameter settings that differed between the two experimental conditions.

Parameter	Active condition value	Passive condition value
Floor factor	2.0	0.5
Interrupt user	true	false
Interrupt self	false	true
Conflict tolerance	N/A	1000 ms
Lapse tolerance	500 ms	4000 ms
Act spacing	50–250 ms	500–1000 ms
Backchannel spacing	2000–4000 ms	4000–6000 ms
Require gaze to seize	false	true

6.3.2 Procedure

In total, there were 30 participants who interacted with Simon in this user study (15 per condition), of which 8 were female (4 per condition). The age range of participants was 17 to 45 years old, with a mean age of 23.5. Ten participants reported experience interacting with young children (5 per condition), such as teaching or babysitting. The participants were recruited from the campus community through mailing lists.

Each participant was randomly assigned one of the conditions and interacted with Simon for two sessions of three minutes each within that behavioral condition. The only difference between the sessions was that the set of objects was changed, and participants were informed that this was the only difference. Participants were told that they should teach Simon about the objects as if he were a young child of about three years of age, but they would not understand what Simon was saying because he would be speaking a foreign language. They were encouraged to talk about properties of the objects, tell stories about interactions between them, or otherwise play with them in a way that was appropriate to a young child.

In addition, participants were primed in ways that constrained their behavior. They were told that Simon could see both their hands, their head, and objects that were on the table, but that if they wanted Simon to attempt to interact physically with any objects, those objects needed to be on the table and their hands could not be covering the objects. This instruction was to prevent users from attempting handoffs to the robot, which were not supported in this study. Finally, participants were instructed to continue engaging the robot and to avoid turning to the experimenter for the entirety of the interaction sessions, even if they were uncertain what they should do.

To trigger the interaction, participants were told to wave to the robot and say "Hello Simon" after the robot's ear lights turned on. For all sessions, the robot started with an uninterruptible greeting turn, comprising a wave gesture with a spoken exclamation. A three-minute timer was started at the end of this turn. At the end of the three minutes, Simon completed his current turn and turned off his ear lights to signal the end of the session.

6.3.3 Measures

6.3.3.1 Post-study questionnaire

After the two interaction sessions were completed, users were asked to fill out a survey with the following questions:

1. How did you find the pacing of the interaction? (slow, medium, fast)
2. Who led the interaction? (Simon, me, about equal)
3. Please rate the following statements about the interaction with Simon. (1 = strongly disagree, 7 = strongly agree)
   (a) Simon was responsive to my actions.
   (b) I had influence on Simon's behavior.

(c) Simon had influence on my behavior.

(d) Simon listened to me.

(e) Simon talked over or interrupted me.

(f) I had to spend time waiting for Simon.

(g) Simon had to spend time waiting for me.

(h) The interaction pace felt natural.

(i) There were silences where nothing happened.

(j) There were overlaps where we both tried to act.

(k) There were awkward moments in the interaction.

4. How would you classify... (1 = strongly introverted, 7 = strongly extroverted)

   (a) ... Simon's personality?

   (b) ... your own personality?

5. (Open-ended) List some adjectives describing Simon's personality.

6. (Open-ended) Please provide a critical review of Simon's social skills.

6.3.3.2 Logged data

Important system events were also logged for each interaction session with timestamps to millisecond precision. These system logs included:

- Petri net events, including transition enables and disables, tokens changing places, tokens changing values, and tokens being spawned or destroyed;
- events for each act being started, paused, resumed, or stopped;
- and reason codes for each full turn or backchannel taken, according to the strategies specified in Section 6.1.3 and Figure 28.

The perceptual data for the human was logged at the framerate of the system, which averaged 30 Hz. This data included pitch and onsets detected from the microphone and all transforms for human skeletons detected from the Kinect. The robot's joint positions were also logged at this rate. A video was taken of each interaction session for future video coding analysis.

6.4 Results

In this section, we present some of the results of our evaluation based on participants' subjective responses, the robot's behavioral data, and participant speech data. We observed that the speech presence signal was extremely robust during the study, but the human skeleton data generally was not (due to arm occlusions and noise). Thus, we can reliably examine human spoken turns, but not human floor-holding that relies on gesture or gaze. The latter requires video coding of the data for accurate analysis, which we leave to future work.

6.4.1 Differences in robot behavior

Our first analysis is a manipulation check that examines whether or not the system's control of these floor regulation parameters actually resulted in different robot behavior across these two conditions. For each modality, Figure 31 compares the fraction of time spent in each behavioral state across all subjects' data for each condition. It can clearly be seen that the active robot spent more time attempting to hold the floor, resulting in taking more full turns; this then led to increased gesturing, speaking, and gazing away from the person's body. In contrast, the passive robot maintained a closer balance between holding the floor and auditing, resulting in more backchannels and gaze at the person's head or hands relative to the active condition.

The fraction of time spent in each of these states differed significantly across conditions for all modality states except for arm gesturing. We also confirmed that the robot did not behave significantly differently across the two interaction sessions

for either condition. Because our subsequent analysis focuses on speech data, we additionally state here that the robot spoke significantly more in the active condition ($M = 87.0$ sec, $SD = 20.1$) than in the passive condition ($M = 39.6$ sec, $SD = 9.4$), $t(13) = 1.30$, $p < .001$.

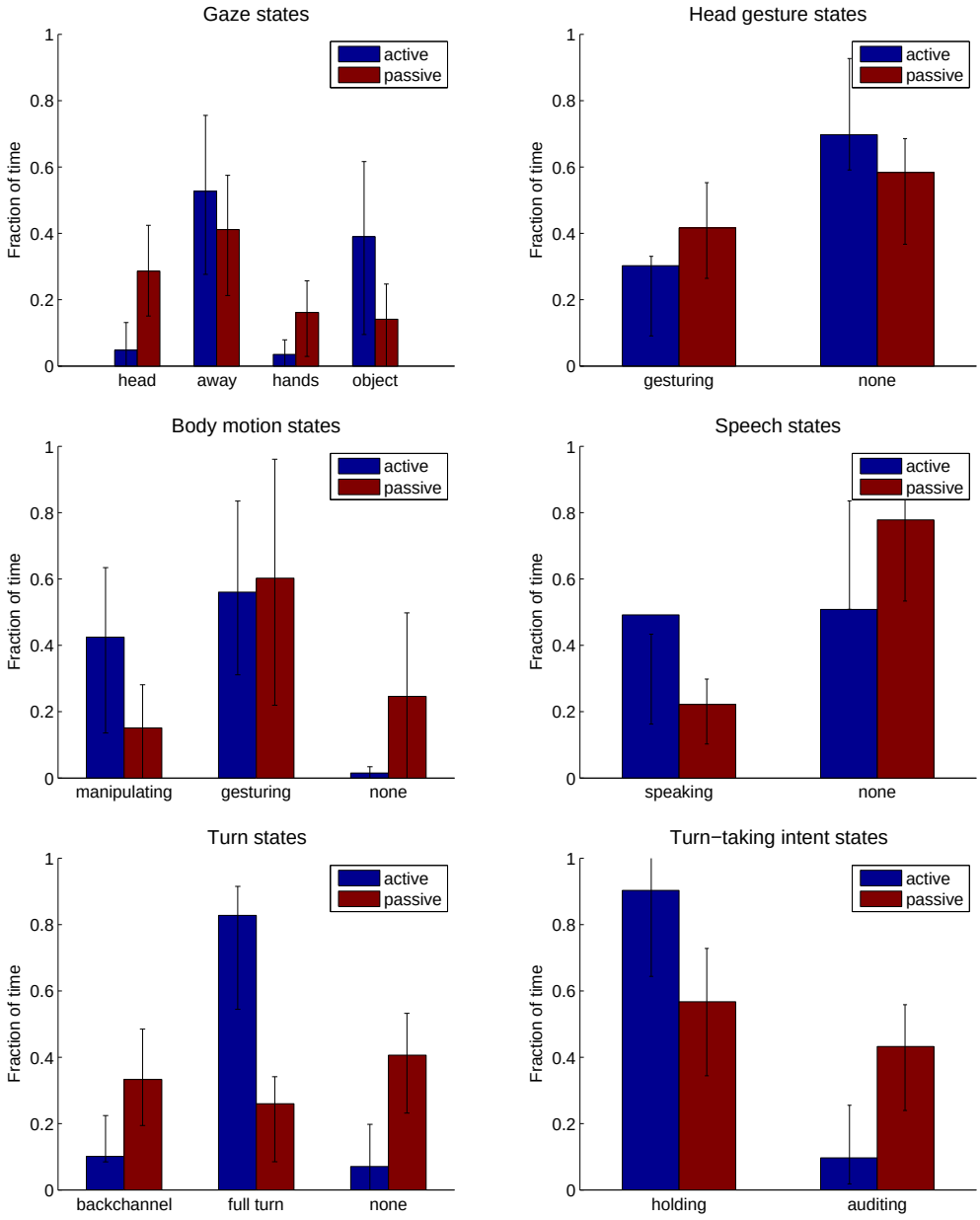


Figure 31: The time that the robot spent in each state, compared across conditions. Each chart shows data for a specific modality. Differences across conditions are significant to $p < .01$ for all modality states except for arm gesturing.

6.4.2 Perception of behavioral differences

Subjective responses indicate that the difference in behavior was perceptible to participants. They were significantly more likely to agree with the statement, *"Simon talked over or interrupted me,"* in the active condition ($M = 6.00$, $SD = 1.07$) than in the passive condition ($M = 4.80$, $SD = 1.21$), $t(14) = 2.96$, $p = .01$. We also found marginal significance for agreement with the statement, *"There were silences where nothing happened,"* where the passive condition reported a higher average value ($M = 4.53$, $SD = 1.64$) than the active condition ($M = 3.40$, $SD = 1.76$), $t(14) = 2.02$, $p = .06$.

Moreover, this difference in robot behavior impacted their perception of the robot's personality. Participants in the active condition perceived Simon as significantly more extroverted ($M = 4.93$, $SD = 1.53$) than participants in the passive condition ($M = 3.46$, $SD = 1.06$), $t(14) = 4.01$, $p = .001$. Subjective reports of participants' own personality introversion ratings did not differ significantly across the conditions. When the participant's self-rating was subtracted from Simon's rating, the average difference was $M = 1.13$, $SD = 1.85$ in the active condition and $M = -1.11$, $SD = 1.67$ in the passive condition, $t(14) = 3.86$, $p = .002$.

Adjectives reported by participants to describe Simon's personality are shown in Tables 5 and 6. Synonyms are grouped in these listings. Table 5 lists all adjectives reported in only one condition, and Table 6 lists all adjectives that were reported in both conditions. In some cases, opposites were reported within the same condition (*extroverted* and *introverted* in the active condition, *responsive* and *unresponsive* in the passive condition, *attentive* and *inattentive* in both conditions), showing the breadth of subjective experience in the study. Although it is difficult to make strong claims about these open-ended responses, these qualitative characterizations overall seem to support the perception of the active robot as more extroverted and dominant.

Overall, these results confirm that the system is capable of manipulating the

Table 5: Adjectives describing the robot's personality that were reported in only one condition.

Adjective	Active	Adjective	Passive
aloof, spacey, distant	3	shy	3
outgoing, extroverted	2	moody, temperamental, flighty	3
gregarious, loud	2	unresponsive, silent	2
bold, confident	2	responsive	1
enthusiastic	1	misunderstood	1
cautious	1	sweet	1
slow	1	naive	1
introverted	1	confused	1
		helpful	1

Table 6: Adjectives describing the robot's personality that were reported in both conditions.

Adjective	Active	Passive
curious, inquisitive	6	6
talkative, rambling	5	3
unattentive, absent-minded, ADD, distracted	3	1
childish, child-like	2	1
attentive, observant	1	2
stubborn, willful	1	2
playful	1	1
contemplative	1	1

robot's initiative and status within an interaction, and that humans can perceive the effects of this manipulation.

6.4.3 Impact on human behavior

We have determined that the robot behaved differently across the conditions and that humans could perceive these differences, but we additionally want to analyze the extent to which the manipulation of floor regulation impacted the behavior of the human. As mentioned previously, we found that we were able to track the human's speaking turns reliably. We analyzed occurrences of robot and user speech across the conditions based on logged data. The starts and ends of user speech segments were determined from the speech presence signal based on a window gap size of 250 milliseconds. Two user logs (one per condition) were generated incorrectly and thus are omitted from this analysis.

In examining this data, we find that participants spoke significantly more in the passive condition ($M = 59.5$ secs, $SD = 26.2$) than in the active condition ($M = 40.1$ secs, $SD = 18.1$), $t(13) = 3.70$, $p = .003$. This is likely due to human aversion to overlapping speech, which led to inhibition of user speech in the active condition but created more opportunities for user speech in the passive condition. In fact, we found that the user spoke slightly but significantly more in the second session in the passive condition ($M = 64.2$ secs, $SD = 28.6$) when compared to the first session ($M = 54.7$ secs, $SD = 23.5$) , $t(13) = 3.72$, $p = .003$, but this was not true for the active condition. This could be explained by the users exhibiting more tentative and uncertain behavior in the first encounter with the robot but taking more control after having seen the robot's passive behavior. As can be expected from these results, the ratio of robot speaking to user speaking also differed significantly across conditions. This ratio[1] was $M = 3.28$, $SD = 3.80$ in the active condition and

[1]Note that these ratios differ from the floor factor parameter setting because this result only takes into account speaking turns, whereas the floor factor ratio also accounts for holding across

$M = 0.85$, $SD = 0.56$ in the passive condition, $t(13) = 4.01$, $p = .001$.

We also discovered that there was significantly more overlapping speech in the active condition ($M = 13.1$ secs, $SD = 7.6$) than in the passive condition ($M = 8.3$ secs, $SD = 4.9$), $t(13) = 3.97$, $p = .002$. We did not analyze whether these overlaps were robot interruptions, user interruptions, or simultaneous starts, since this requires more contextual knowledge and would need to be determined through video coding.

Given that the user spoke more to a passive robot than an active one, we hypothesized that there may have been regularities in the robot's modality-specific actions that encouraged the user to seize the floor. We compared robot behavioral states at the starts of user speech turns to the robot's overall modality-specific behavior state distributions for each condition and found that this hypothesis was not supported by the data. Figure 32 shows the high similarity between the distributions for the active condition (data for the passive condition is even more similar, and thus is not provided). An exception is the speech modality, where the absence of robot speech favors a user seize attempt. Hence, in this data set featuring open-ended human-robot turn-taking, a simple "nod and a glance" [23] does not suffice for controlling or predicting when users will take turns.

6.5 Discussion

Results indicate that CADENCE is able to manipulate social dynamics through variations in floor regulation behavior. Experimental manipulation of our model parameters resulted in significantly different robot behavior that caused participants in different conditions to attribute significantly different personality types to the robot. Importantly, this manipulation in robot behavior succeeded in eliciting different behavior from the human partner across the two groups, changing the social dynamics of the dyad. The human partner acted longer and more often when interacting with

other modalities.

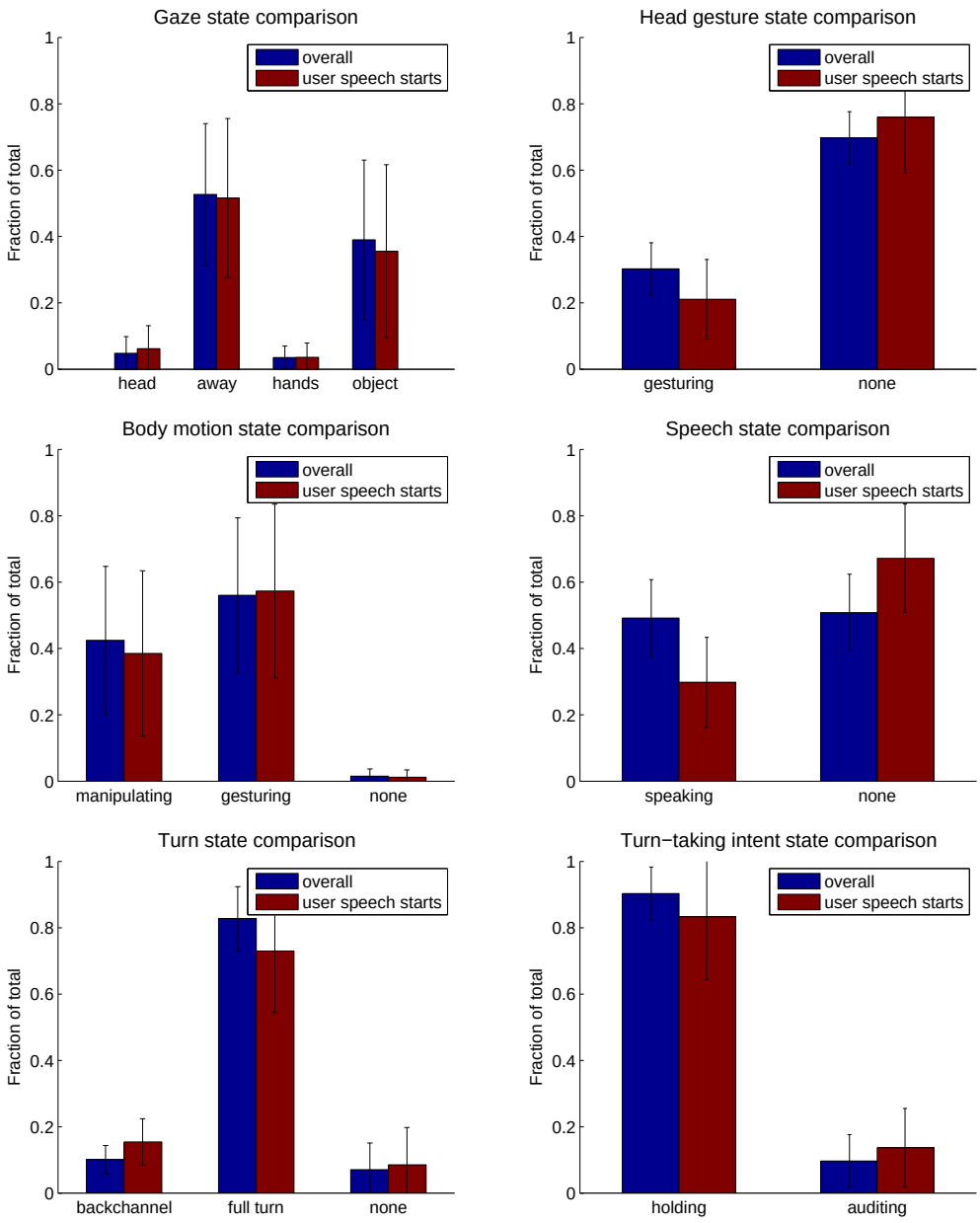


Figure 32: This data is taken from the active condition only. The figures compare distributions of robot behavior data at the times of user speech starts to the overall distribution for the active condition. Overall, the distributions appear highly similar, making it difficult to predict onsets of user speech. The most substantial difference can be seen in the speech state comparison.

a passive robot than with an active one.

Here we make several additional observations about challenges for future work.

6.5.1 Improvements to backchanneling

In general, there were fewer significant differences in subjective ratings across conditions than we had expected. In both conditions, subjects thought the interaction was slow to medium in pacing, about equal in who was leading, and had overlaps in turns. Subjects also did not consider Simon more of a listener in the passive condition. In examining the data in conjunction with anecdotal observations of the study, it seems that spoken backchannels did not function as supportive auditing behavior as much as hypothesized. This may explain why several participants perceived Simon as *"talkative"* even in the passive condition (Table 6); users may have perceived Simon's backchanneling as taking floor time, albeit in short spurts. In light of this, it may have been more appropriate for backchannels to count against the robot's floor time after all, but perhaps with reduced weight.

Improving timing and comprehensibility of backchannels seems necessary for generating more successful auditing behavior. Some part of the lack of accessibility of backchannels in this study can be attributed to the artificial language. When using true linguistic backchannels in a task context, it will be important to consider the different functional types of backchannels and the information they convey [7]. In addition, backchannel timing is a sensitive issue. Backchanneling periodically to maintain engagement throughout a lapse could be inappropriate if the user is thinking silently; such repeated backchannels could be perceived as disruptive or annoying. On the other hand, it can be completely appropriate to backchannel with quick spacing in response to short, high-information utterances from the human. More modeling of these types of information exchanges may be needed for the robot to be able to time backchannels appropriately.

6.5.2 Modality-specific bottlenecks

One simplification we made in our current implementation was to treat the conversational floor as a singular resource to be negotiated by the two parties. This resulted in action across all modalities being combined to classify holding behavior. Realistically, interaction dynamics are also strongly defined by modality-specific bottlenecks. For example, overlapping speech is avoided, as well as close proximity that can lead to physical collision or uncomfortable social distance, but cross-modality simultaneity such as speech from one party and gesturing from another does not necessarily constitute a conflict. On the other hand, correlation of actions would still be expected across modalities due to the innate structure of information bottlenecking in the turn-taking interaction—for example, in the way that gaze behavior accompanies speech turns, and in the way that speech is synchronized with deictic gestures [75].

A next challenge is to define the roles of these modality-specific bottlenecks more clearly when planning higher-level cross-modality turns. For example, seizing the floor to speak may inhibit the partner's tendency to speak but does not necessarily require that the partner stop acting in the workspace. Similarly, taking a turn using a particular object prohibits the partner from using that object but does not prohibit him from taking a speaking turn or from performing an action with another object in the workspace. Indeed, there were many instances in the study of a user speaking over Simon's manipulation actions (e.g. providing words of encouragement or semantic object labels) that were socially appropriate and non-conflicting. On the other hand, rigidly enforcing complete independence of individual modalities is similarly oversimplifying, as gaze and body motions do influence verbal expression and suppression; this explains why telephone conversation dynamics differ so significantly from face-to-face dynamics [109]. Modeling appropriate constraints between the floor resource and modality-specific resources will be necessary in order to capture the fluid human-robot interaction that we desire from our system. This is the principle

limitation that is addressed by the changes in Chapter 7, and evaluated in Chapter 9.

6.5.3 Contextual parameter setting

Our study demonstrated only two possible parameter settings of the system. We emphasize again that we do not universally advocate one of these conditions over the other, but simply state that our model allows the robot to exert some control over social dynamics and influence who takes more initiative in the interaction. The appropriate parameter setting should ultimately be dictated by social context. A point of interest for our future work is such a contextual setting of these parameters by recognizing detectable and generalizable characteristics of a social situation or an interaction partner.

As an example, a tour guide robot may benefit from "active" floor regulation parameters while leading a crowd, whereas a butler robot may need to rely on "passive" parameter settings when serving its owner. We believe that the system parameters are intuitive, and as such can be set to constants defined by a robot designer to suit a particular application or culture; or perhaps a personal service robot can be set to use custom parameters preferred by its owner. We are also interested in further exploring appropriate turn-taking behavior for a robot learning from a human teacher; a spectrum of passive to active behavior is possible for a robot active learner, not all of which is actually conducive to successful learning [20].

In addition to static parameter settings based on a task domain, it may be useful to modulate the parameters dynamically throughout an interaction. This technique could be used to support status-elevating and status-lowering transactions [52]. In addition, such modulation could allow better adaptation of the robot to the human. It is known that people gradually synchronize the timing of communicative behaviors to interaction partners over time [19], and we have also observed such convergence

in our previous work [25]; this capability could potentially improve a dyad's fluency. It may also be useful to consider adapting to the human's affective state, if it is perceivable through cues like vocal prosody. A stressed or angered human may desire different turn-taking dynamics than a relaxed one. Conversely, the control of the robot's turn-taking behavior could be modulated by an emotion model, serving a communicative purpose regarding the robot's internal state.

6.6 Summary

In this chapter, we have described a turn-taking model-controller that is intuitively parametrized to allow the robot to achieve a range of different social dynamics, which can be altered to target specific interaction scenarios. In applying contrasting parameter settings of CADENCE in a user study, we have demonstrated that: (1) manipulating these floor regulation parameters results in significantly different robot behavior; (2) people are able to perceive this difference, as well as attribute different personality types to the robot; and (3) changing the robot's personality results in different behavior from the human, manipulating the social dynamics of the dyad.

Our results confirm the utility of CADENCE for controlling turn-taking dynamics but also point to shortcomings specifically related to backchannel communication and the appropriate modeling of modality-specific bottlenecks. Specifically, backchannels are "short turns" whose semantic function cannot be ignored, and modalities must on occasion be treated differently when deciding to interrupt behavior. These are precisely the limitations we seek to address in the next two chapters.

CHAPTER VII

MULTIMODAL RESOURCES

Why aren't all actions executed simultaneously and instantaneously in real life? The general timing of situated interaction can be interpreted as being due to resource limitations. We claim that these resource limitations fall into at least two categories of being *physical* or *cognitive*. Occupying a physical resource is straightforward to envision: if a person's hand is in a particular location or using a particular tool, this prevents the robot from being in that location or using that tool within the duration of that action. The prototypical cognitive bottleneck is the speaking floor. This bottleneck is due to a human attentional resource limitation defined by Baddeley as the phonological loop [6], which describes auditory processing and generation as using the same buffer. Because sound source separation and parallel semantic processing are cognitively difficult, social custom dictates that we take turns to speak and has also evolved in many cultures to indicate that we don't want to listen if we are currently speaking.

In addition, for a robot capable of action in multiple social modalities, there can be ambiguity as to which modality or combination of modalities to use. Modalities can be functionally redundant (e.g. saying hello, versus waving to a person, versus doing both together), although this redundancy only results in acceptable behavior if timing is appropriately coordinated between modalities or a modality is used singularly. We argue that resource availability is a factor in this decision.

Previous systems, including the version of CADENCE described in Chapter 6, have assumed a unitary notion of the floor. None properly handle multiple types of resources concurrently across multiple modalities, which we believe is essential for

achieving fluent human-robot joint action. The foundation of this iteration of the system is the process for resource monitoring, in which shared resources of a given type are formulated as mutual exclusions. We describe how this resource monitor interacts with interruptible action processes that seize and yield resources, and how we formulate actions so that they can be dispatched to these processes in a semantically synchronized fashion. The result is appropriate multimodal turn-taking, which we evaluate in the experiment in Chapter 9.

7.1 Resource monitoring

Resources are required to execute actions, and using a resource for execution should suppress actions from other processes that require that same resource. This competition can exist between internal robot processes or between the robot and an external process (e.g., a human partner, or the environment). An example of internal competition is control over robot degrees of freedom; a social robot cannot simultaneously move neck DOFs for both gaze and gesture. Competition with external processes requires turn-taking. Our system considers multiple types of resources: objects, spatial regions, the speaking floor, and robot DOFs. These resources are used by actions in multiple modalities. For example, picking up an object uses the object, spatial regions, and arm and hand DOFs.

Given the set R_x of all resources of type x, we define components of resource monitoring as follows. Each resource is represented as a token whose ownership is transferred through the graph. Available resources are owned by the global place p_{free}. Competition is defined as being between a set of resource controllers I, each comprising places p_{owned_i} and $p_{requested_i}$. The following invariants hold between controllers:

$$
\begin{aligned}
|R_x| &= |p_{free}| + \sum_i |p_{owned_i}| \\
\forall i \in I, p_{requested_i} &\subseteq R_x
\end{aligned}
\tag{4}
$$

Each controller has the ability to request, seize, and release resources. For each controller, the internal firing mechanics are defined as follows:

$$
\begin{aligned}
A &: A \subseteq R_x - p_{requested_i} \\
B &: B \subseteq p_{requested_i} \setminus p_{owned_i} \\
C &: C \subseteq p_{owned_i} \setminus A
\end{aligned}
\tag{5}
$$

$$
F_i = \begin{bmatrix}
 & t_{request_i} & t_{seize_i} & t_{release_i} \\
p_{owned_i} & \varnothing & +B & -C \\
p_{requested_i} & +A & \varnothing & \varnothing \\
p_{free} & \varnothing & -B & +C
\end{bmatrix}
\tag{6}
$$

This model enforces a mutual exclusion between p_{free} and $p_{owned_i} \forall i$, while allowing multiple processes to put in requests for resources simultaneously. We define the initial marking as all known resources being available and no resources being requested:

$$
\begin{aligned}
\forall r \in R_x, r &\in p_{free} \\
\forall i \in I, p_{requested_i} &= \varnothing
\end{aligned}
\tag{7}
$$

The model just described can be directly used for internal processes, i.e. DOF control. Figure 33 depicts how two resource controllers are connected together for turn-taking control. Two transitions are added, t_{yield} and $t_{bargein}$, take into account the state of the user in order to decide whether to yield the resource or barge in and take ownership directly.

Figure 33 shows the turn-taking transitions from the perspective of the robot and depicts a feasible marking. $|R_x| = 2$, with one resource being available and one being owned by the robot. Both the robot and the user are requesting more resources than they currently own. If the user's requested resource is the one owned by the robot,

the robot may yield it to the user. The robot additionally needs the available resource in p_{free}. This diagram could depict a real-world state of the robot holding on to an object, with another object being on the table. The robot needs both objects, and the user needs one of them.

On the robot side, the resource controllers interface with action-handling processes. Actions that are intended result in requesting resources, and resources must be seized and owned in order for actions to execute successfully. Resources are released when actions complete. In contrast, transitions for the user controller are implemented as perceptual processes. For example, an object is requested when the user is perceived to be reaching towards it and owned when the user has it in hand, and the speaking floor is owned based on voice activity detection. Subtler cues such as gaze, prosody, and discourse markers are also applicable. Additional timing parameters can throttle or trigger seizing and yielding, as in [27].

7.2 Resource-aware action execution

Previously, we described interruptible action processes in Chapters 5 and 6. However, a key limitation of those versions was that only a unitary resource was modeled, the floor, which can result in undesired cross-modality interruptions. Such interruptions can be heavy-handed and inappropriate. For example, an interruption in the speaking modality should not necessarily interrupt a manipulation action.

It is difficult to account for all possible situations in which such cross-modality interruptions are appropriate or not appropriate, because these can be dependent on the semantic content of turns. As an example, a human may say to a manipulating robot, "Good!" or "Stop!" Most conservatively, we assert that an interruption is at least appropriate if a resource that is required for execution is lost during execution. That is, if a robot is reaching for an object, resource loss of the speaking floor should not *necessarily* demand an interruption, but resource loss of that object should.

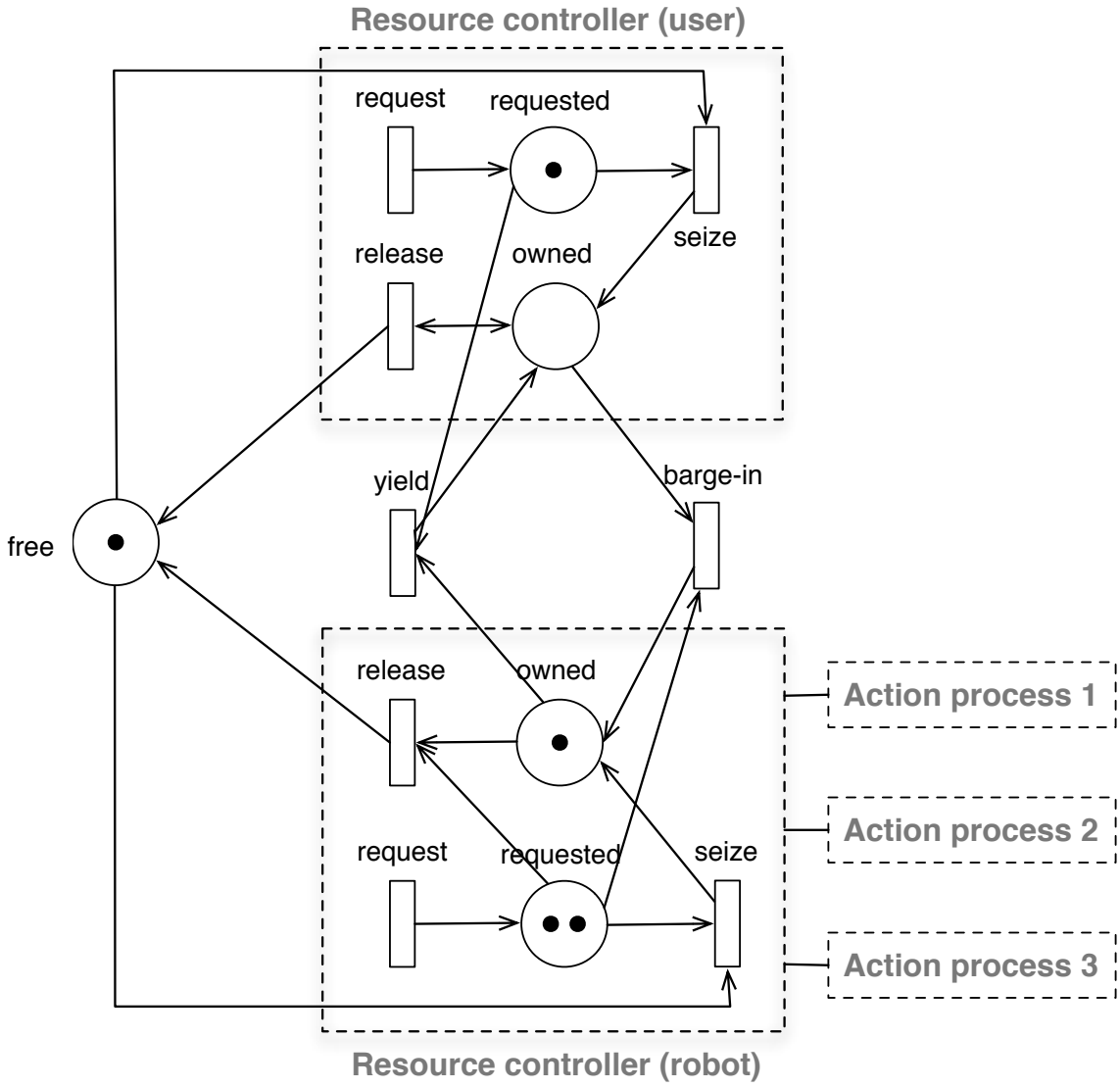


Figure 33: A resource monitoring model for one type of resource, shown from the perspective of the robot. A separate such model exists for each resource type.

To achieve this behavior, we connect our previous notion of interruptible action processes to the resource model described in Section 7.1. The action process template is shown in Figure 34 and reflects the model of execution presented in Chapter 6 that includes hesitating before interrupting. Such behaviors are automatically generated with respect to resource losses that are incurred during execution. Table 7 explains the key interface points between the resource model in Figure 33 and the action process model from Figure 34. Note that interfaces to multiple resource controllers may exist even though the diagrams show connections between only a single resource controller and a single action process. For example, manipulation actions may require objects, spatial regions, and robot degrees of freedom.

In our framework, an action is a semantic entity specified independently of the action execution process that handles it. For example, a pointing gesture might be accomplished through motion planning, animation playback, or joint space interpolation. An action $Act(E, R_{pre}, R_{post})$ is a function of a set of semantic entities E, resource preconditions R_{pre}, and resource postconditions R_{post}. For example: $Grasp(\{object_i, hand_j\}, \{hand_j{}^+\}, \{object_i{}^+, hand_j{}^-\})$ is used to grasp an object, where $hand_j$ represents a DOF set for a hand. $R_{pre} = \{hand_j{}^+\}$ indicates that the hand DOF resources must be owned before execution can occur, and $R_{post} = \{object_i{}^+, hand_j{}^-\}$ indicates that after execution, control over the hand DOFs is released, and control over the object persists.

Referring again to Figure 34, such an action is queued for execution in its handling process and starts only when resource prerequisites hold. Transition t_{start} then starts the action and places the action token in $p_{started}$. The transition t_{hold} continually monitors the action execution for its progress, state of intention, and resource availability. If no resource conflict or intentional change occurs, the action finishes normally, firing directly from $p_{started}$ through t_{finish} to $p_{finished}$. If the action is no

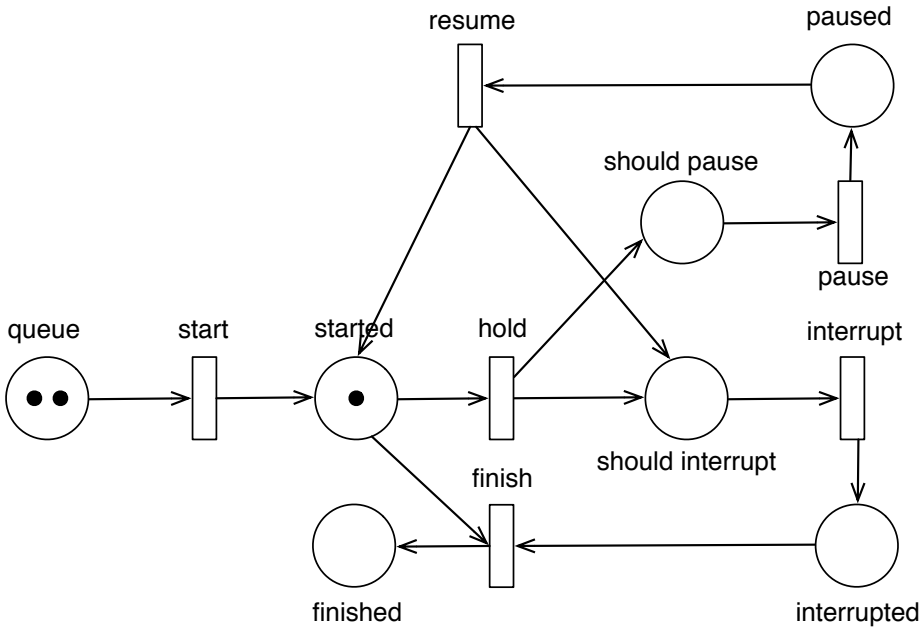
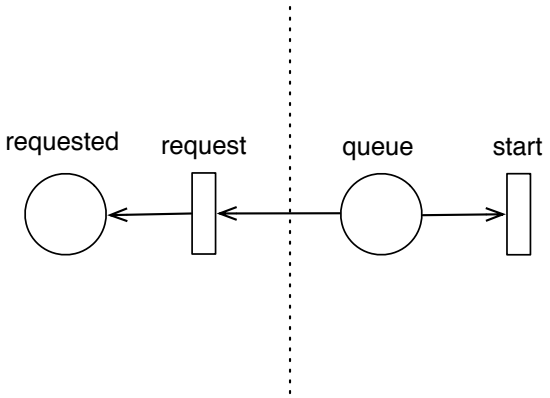


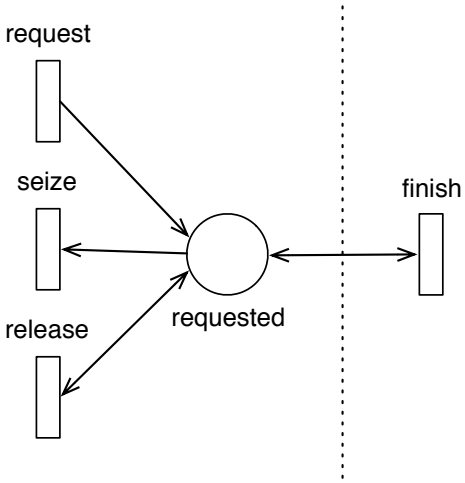
Figure 34: Each modality's action process follows this template, with minor differences. An interruptible action process includes the ability to pause in the presence of a potential resource conflict and decide within a deadline whether to resume or interrupt the action.

longer intended, t_{hold} moves the action token to $p_{should-interrupt}$ to interrupt the action. In the presence of a resource conflict, t_{hold} moves the action token to $t_{should-pause}$, causing t_{pause} to fire. The transition t_{pause} generates hesitation behavior to bide time while observing resource availability changes. A hesitation is both functional in the face of uncertainty (by balancing the cost between continuing and stopping) and also displays a contingent communicative signal [76]. So as not to deadlock, t_{resume} must decide within a deadline whether to resume execution or to interrupt the action fully in order to yield the resource.

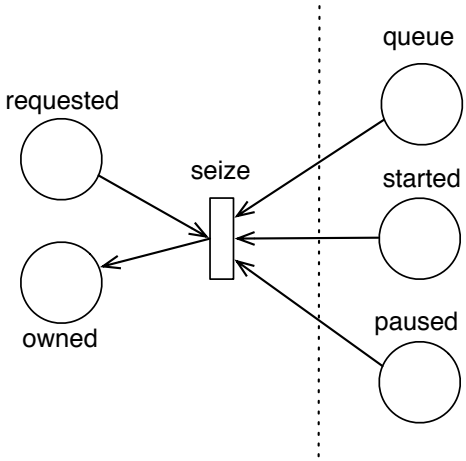
Algorithm 1 summarizes the functionality of the transitions in the action process template. These firing functions produce general-purpose resource-aware behavior. Each modality must then provide custom behavior for Start, Hold, Pause, Resume, Stop, and Finished for a specific Act type.



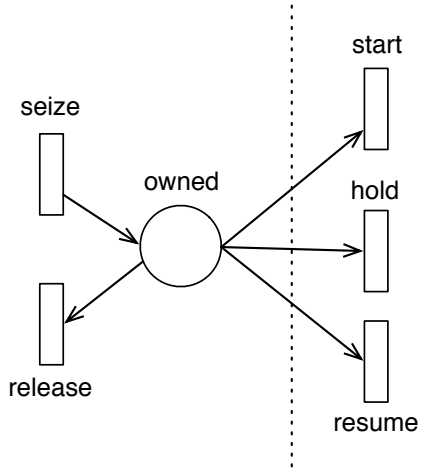
When action tokens are queued into p_{queue}, $t_{request}$ places resource tokens in $p_{requested}$ that are needed for that action.



$p_{requested}$ contains resource tokens that are needed for intended actions. When the action finishes, t_{finish} updates $p_{requested}$ by removing resource tokens that are no longer required.



t_{seize} will seize requested resources if possible, and place resource tokens in p_{owned}. t_{seize} operates whenever actions are intended — queued, started, or paused. Resources must be owned to make progress on an action.



When p_{owned} contains resource tokens, the action can make progress. The action can start running through t_{start} and is monitored through t_{hold}. If the action was paused due to a resource loss but then reacquired, t_{resume} will restart the action.

Table 7: Each diagram depicts an interface point between a resource controller and an action process. Nodes left of the dotted line are part of the resource controller, and nodes right of the dotted line are part of an action process.

Algorithm 1 Pseudocode for transition firing functions within the interruptible action process template.

```
function FIRE(t_start,Act)
    if resources(Act) ∈ p_owned then
        START(Act)
        move Act token to p_started
    end if
end function

function FIRE(t_hold,Act)
    update Act goal monitors
    HOLD(Act)
    if interrupt self = true then
        if Act is intended then
            if resources(Act) ∉ p_owned then
                move Act token to p_should−pause
            end if
        else
            move Act token to p_should−interrupt
        end if
    end if
end function

function FIRE(t_pause, Act)
    PAUSE(Act)
    move Act token to p_paused
end function

function FIRE(t_resume,Act)
    if timePaused(Act) > maxWait then
        move Act token to p_should−interrupt
    else
        if timePaused(Act) > minWait and resources(Act) ∈ p_owned then
            move Act token to p_started
        end if
    end if
end function

function FIRE(t_interrupt,Act)
    STOP(Act)
    move Act token to p_interrupted
end function

function FIRE(t_finish,Act)
    if FINISHED(Act) then
        move Act token to p_finished
    end if
end function
```

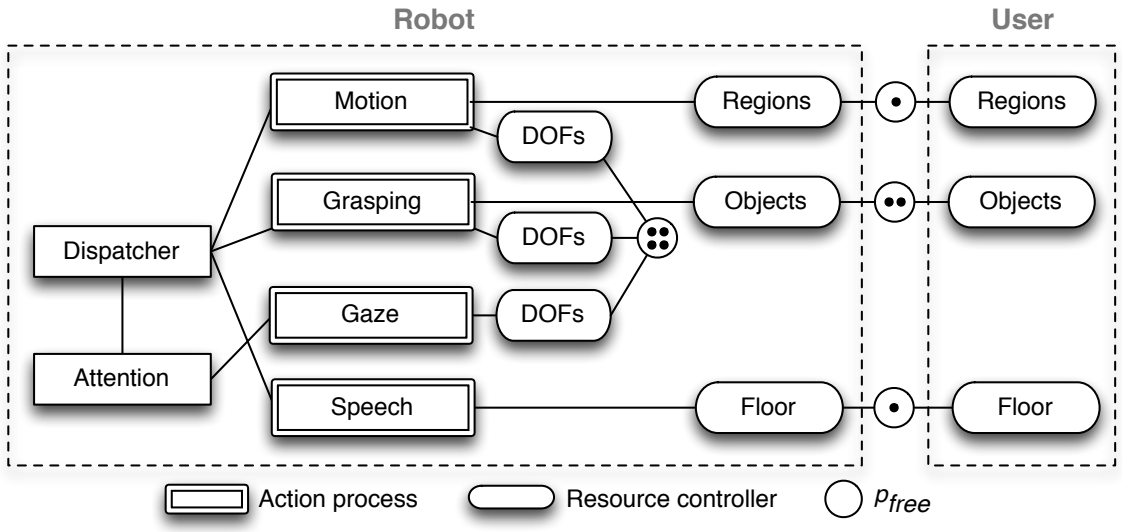


Figure 35: The interdependencies between processes in the system. Each component is a TPN in itself. Turn-taking occurs between resource controllers of the same type.

7.3 Multimodal action alignment

Later in Section 8.3 (on page 135), we will be defining a grammar that is used to bridge semantics, surface realizations, and actions. For now, it is sufficient to describe the grammar as hierarchies of nodes to which actions are attached. Following the structure of an AND-OR tree, nodes can be disjunctive, indicating that their children are to be treated as alternatives. This grammar also specifies temporal constraints for action alignment as follows. Each node N_x is associated with an interval $I_x : [t_{Sx}, t_{Ex}]$ for starting and ending times. If $|children(N_x)| = Y$, then for all nodes $N_y \in children(N_x)$ where $y \in Y$, $t_{Sy} \geq t_{Sx}$ and $\bigcup_y I_y \subseteq I_x$. That is, child nodes are always executed after their parents. In addition, $t_{Ey} <= t_{Sy+1}$, indicating that nodes are always executed after any preceding siblings. A node is considered completed when all of its actions and its children's actions are finished. The exception is the notion of disjunctive nodes, for which only a single child's actions are executed. This representation can be used directly to define a temporal constraint satisfaction problem for scheduling resources to be used by actions.

The temporal constraints of the grammar are shown in Figure 36, which has a

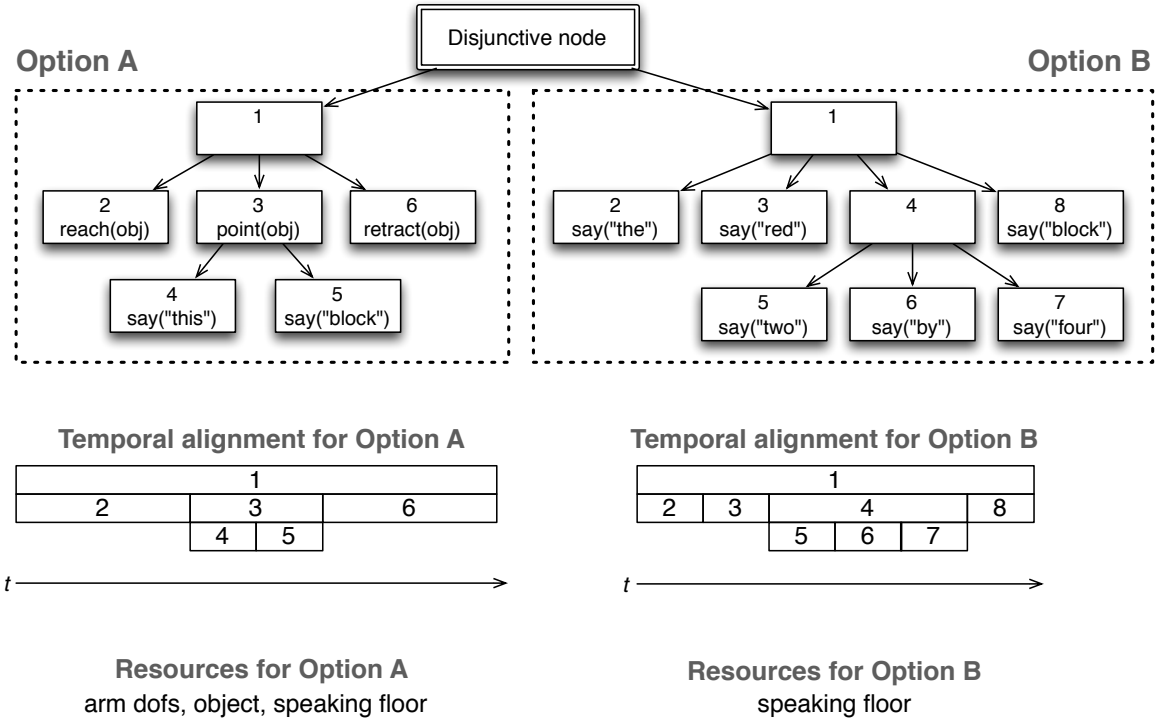


Figure 36: Temporal constraints between parents and siblings within action hierarchies define the alignment of when actions can start and stop. Shown are two options for generating an object referring expression, which use different resource sets.

disjunctive root node. The figure shows an example of having two options of modality combinations for generating a referring expression. On the left subtree, the robot uses speech and gesture by pointing and speaking a more compact phrase using "this" as a determiner. On the right subtree, the robot uses speech only. Factors relevant to the situationally appropriate selection of modalities include efficiency, cognitive demand, risk of interpretational error, and resource availability. The work in this chapter specifically addresses the aspect of resource availability.

Figure 37 shows how the dispatcher serves as the interface between action hierarchies generated by the domain model, and the action execution processes in the behavior TPN.

The incremental nature of the execution also allows for a simple implementation of top-down gaze generation by which the most recently visited semantic entity is the focus of attention. This contrasts with bottom-up saliency models, which apply

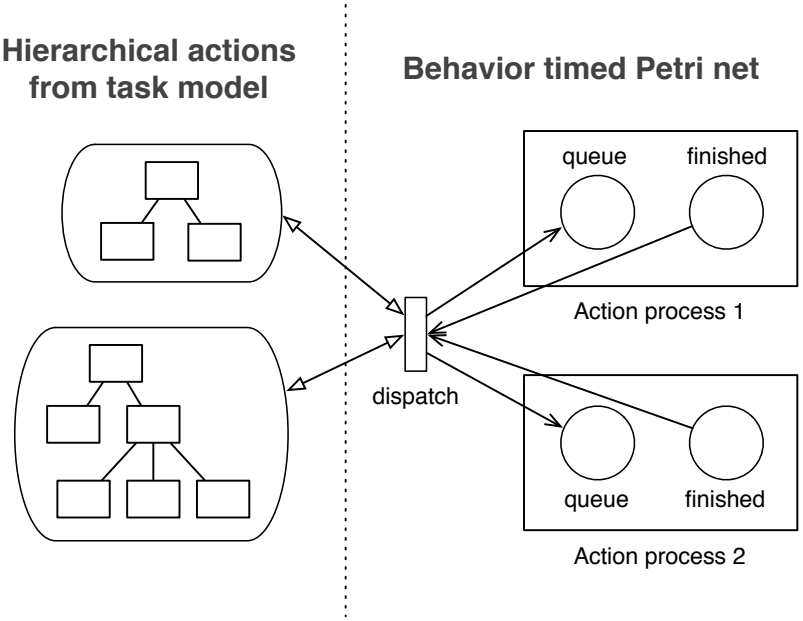


Figure 37: The dispatcher starts actions based on the temporal constraints described in Section 7.3. Action tokens are placed in p_{queue} for their corresponding action processes. When an action is completed, its token reaches $p_{finished}$. $t_{dispatch}$ consumes the token and proceeds with dispatching subsequent actions.

higher weight to important perceptual events such as motion and loud sounds. Purely bottom-up models can make the robot seem distractable when engaged in a task with a user. We believe that both types of models are necessary for social robot cognition.

7.4 Scheduling

A collaboration act is represented for execution by a hierarchy of grammatical nodes, which have lower-level action primitives attached. When multiple such actions are parallelized, it can become possible to schedule actions for oneself that result in resource contention. An example is both arms needing to access the same spatial region. This can lead to strange behavior like self-interruptions. In the worst case, the robot can deadlock itself.

A simple way to avoid such behavior is to collect all resources from an intention hierarchy, and never simultaneously execute two intentions that share resources. This can result in suboptimal execution. For example, an intention for a speech act

might represent a choice of speaking only versus speaking and pointing. The pointing subhierarchy can be performed by the left or the right arm. Collecting all of these resources together prevents the robot from speaking while performing other actions with its arms even when such a combination is feasible.

A solution to this is to collect the *disjunctive* resource sets for each intentional hierarchy. In the aforementioned example, the disjunctive resource sets would be: 1) floor only, 2) floor + right arm, and 3) floor + left arm. Using this representation, we can use the scheduling framework Tercio to compute a schedule [41]. Tercio computes near-optimal schedules for simple temporal problems by iteratively alternating resource allocation and action sequencing steps. We define temporal constraints for all of the within-hierarchy dependencies described in Section 7.3 to be used in the sequencing step. We can also define between-hierarchy dependencies based on turn-taking parameters such as act spacing or response delays. The schedule is updated whenever resource availability changes in a way that invalidates the current schedule or when new intention are queued.

7.5 Limitations and next steps

Several direct extensions are immediately apparent. One is the addition of more resource types. An example is visual attention, which would require monitoring the human's head orientation and would be a required resource for any deictic gestures. Other extensions would be implementing more sophisticated resource signaling models. For example, more subtle signals for spoken turn-taking include mouth-opening or aspiration, and more explicit signals for physical turn-taking include statements like "go ahead."

One current design limitation of the system is that there is no support for concurrent ownership of resources. Thus, this model is not at all suited for tasks requiring such shared ownership, such as both the human and the robot carrying an object

together. The system also currently does not model handovers.

A greater question is how semantics should influence resource usage. In Chapter 3, we saw that minimum necessary information was critical to the way that humans yielded resources. If the relationship between resources and task semantics cannot be abstracted in a general way, then there is little purpose to having resource-controlling processes, since they must encode domain-specific rules. In the next chapter, we design a dialogue system that supports semantically rich situated interactions while abstracting general-purpose turn-taking behavior away from those same semantics.

CHAPTER VIII

DIALOGUE FOR SITUATED COLLABORATION

In some ways, modern interaction research promotes a view of behavior being divorced from cognition. Interaction "designers" are responsible for ghostwriting the robot's utterances, generating canned animations for playback, and packaging them together into black-box states. Mostly, the robot is the designer's puppet, parroting back a script. Because these representations are so domain-specific, it is subsequently claimed that the only generalizable outcomes from these interactions are "principles" or "guidelines" that influence the process – essentially fancified advice for future interaction designers. Of course, these principles augment the designer rather than the robot.

What if the robot could instead model the process in the designer's head, and act autonomously from its own knowledge and perception? This is the premise upon which we build the system in this chapter. We consider the scenario of *situated collaboration*: two agents embedded in the physical world, cooperating to reach a shared goal. When they interact, they must share resources, integrate information from the world and their partner, dispute suspect ideas, alter their beliefs, and reach common ground.

In this chapter, we describe a system for robots to have collaborative dialogues with humans. The system we develop synthesizes ideas from linguistic theories about how humans have conversations, model mental states, build common ground, and repair misunderstandings. In contrast to the previous chapter, which was focused on *action*, this section is primarily concerned with *meaning*. We describe how we modularize domain-specific knowledge and design general-purpose processors that

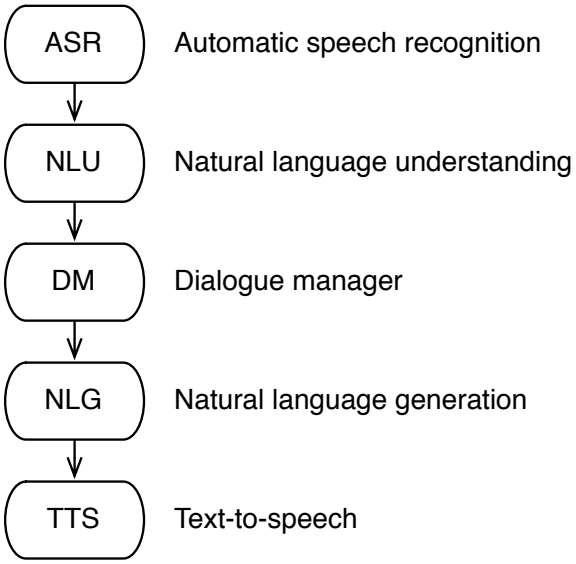


Figure 38: Traditional dialogue systems model only speech

interpret and generate dialogue acts based on that knowledge. We end this chapter with some ways that the structure and semantics of dialogue influence the timing of turn-taking.

8.1 Dialogue with robots

Dialogue systems are most commonly speech-only systems. Figure 38 depicts how such dialogue systems are typically conceptualized. In the dialogue manager slot goes any number of approaches: finite-state machines (FSMs) [89], partially observable Markov decision processes (POMDPs) [125], probabilistic rules [68], form-filling [40], dialogue trees [14], and so on.

A social robot lends more complexity to this view. Its sensory input includes not only ASR, but also visual information about the surrounding environment and the human's physical activity. It generates not only speech but arm motions and gaze directions. Figure 39 depicts a system diagram for situated collaboration. We will revisit this figure at the end of this chapter after having explained all the components.

In a situated setting, the need for integrated understanding and generation across

modalities heavily favors dialogue approaches that rely on deep semantic understanding, in contrast to data-driven approaches that learn how acts can be probabilistically sequenced. When text is the only surface form and actions are never concurrent, it can be convenient to abstract away entire turns as single symbols, simplifying the problem of dialogue to moves in a game [127]. For a physically situated agent, however, actions are temporally extended and reasoning must be bridged across modalities. An object referring expression may coordinate physical action (pointing with an arm), visual attention (gaze), and speech that depends on surrounding context. These can be simply bound together by an understanding of the object's location and other visual attributes. Data-driven approaches fall short in their lack of scalability in handling combinatorial situations and alignments.

Previous approaches to building dialogue systems with deep semantic understanding have been successfully demonstrated, as in Allen's PLOW system [1] that relies on the semantic TRIPS parser [37], and in the robot knowledge-acquisition dialogues

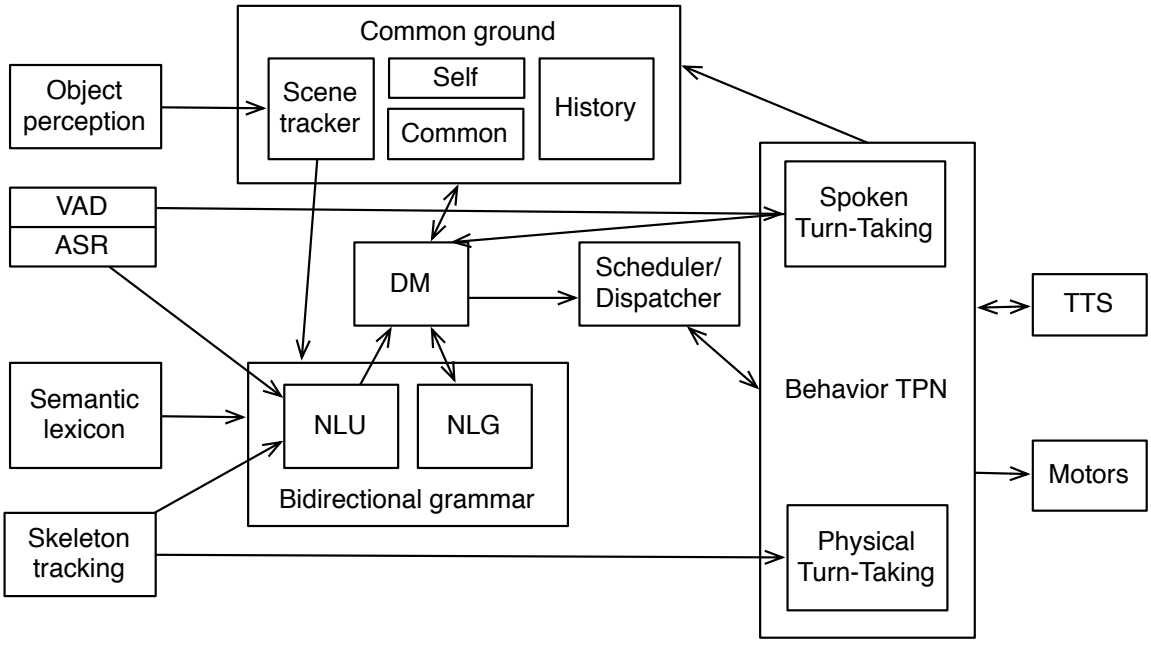


Figure 39: Dependencies of components used for situated collaboration. For comparison and abbreviations, refer to the speech-only system diagram in Figure 38.

of Kruijff et al. [61], Lemaignan et al. [67], and She et al. [105]. The popularity of data-driven approaches, e.g. [125], capitalizes on simplifying assumptions that can be made in speech systems that do not occur in a situated environment, such as call center dialogues. The work in this chapter builds on prior work in deep semantic collaboration, as well as other models of collaborative intention such as SharedPlans [43], in order to extend these techniques for controlling the timing of temporally extended actions within the context of a situated collaboration. Notably, much prior work has provided extensive treatment of collaborations that focus on the learning or execution of steps in a task *plan*. In this section, we focus on the additional problem of agents shaping and converging on the concept of a mutually agreed-upon task *goal*, from which physical actions can then be spontaneously derived.

The dialogue system in CADENCE is designed around the idea of composable semantic entities. It contains the following components:

- The *semantic lexicon* holds semantic units of domain-specific knowledge.
- The *cognitive grammar* is the bidirectional compositional system for interpreting semantics from the natural language surface form and generating natural language from semantics.
- These semantics are further organized into *records* and *propositions* at the top level.
- Information state used for dialogue is tracked in *common ground*.
- *Collaboration act (CA) processors* act on common ground state to generate *collaboration acts* to execute.

8.2 Semantic lexicon

The semantic lexicon describes all of the units of knowledge that the robot can reason about in an interaction context. We refer to these primitives as *semantic units*. These

are not simply symbols that can be strung along into grammatical sequences of text (e.g. "colorless green ideas sleep furiously"), but are meaningful in their relationships to each other. We refer to Fillmore's theory of frame semantics for the idea that words are not understood in isolation but in terms of their relations to other concepts within a scope of essential knowledge called a "frame" [38]. The concept of a buyer cannot be understood absent of sellers, goods, and currency; the concept of "pick" cannot be understood without an object and a hand.

Precisely how much content should constitute a semantic unit? For this, we refer to the Reprise Content Hypothesis, the strong form of which is stated below:

> A nominal fragment reprise question queries exactly the standard semantic content of the fragment being reprised [39].

Put another way, the fragments that are used to clarify and repair understanding in interaction give a clue as to how human minds organize conceptual knowledge. A practical interpretation would be to say that, given a specific interaction context, semantic units should represent quantities of information equivalent to reprise fragments for them in that interaction, and need be no smaller. Clark and Krych have previously highlighted that conversation is a process of continuous understanding [30], a principle that guides the development of spoken dialogue systems based on streams of incremental units [101]. The Reprise Content Hypothesis provides a recommendation for exactly how small such incremental units need to be.

For semantic entities to be talked about, they must be associated with words and grammatical function that can produce their *surface realization*. Each unit also provides a wh-form so that they may used in interrogative acts. Semantic entities can be installed without grammatical attachments, in which case they are still reasoned about but never referred to.

8.2.1 Primitives

Semantic units in CADENCE include:

A *feature* is an attribute that implies a domain of *feature values*. Features can be numerical or discrete. Feature values are typed to a specific feature and are typically adjective phrases, adverb phrases, or noun phrases. Features provide a verb form, and a wh-form if they are to be used for feature queries (e.g. "where" for location, "how wide" for width). Features are not necessarily just attributes but can represent other relationships as well, such as meronym/holonymy.

An *object* is a bag of feature values. The maximal set of features for an object type is given by its *object factory*. For example, a block object could have feature values for color, shape, and location. The feature values are typically adjective or adverb phrases, and the object factory provides ordering rules for them. Objects with location values can be referred to through pointing and gazing. When parsing and generating referring expressions for objects, the situated context is taken into account. In addition, the pronoun "it" can be used to refer to recently referred objects in dialogue history.

A *category* is an object factory combined with a set of features that is a subset of that object factory. An example of a category is "red blocks." The subset can be the empty set, e.g. "blocks."

An *action* may be parameterized on multiple objects or feature values. Examples are look(object), pick(object), and place(object, location).

An *agent* is a subclass of objects in that it has feature values like location, but it can also execute actions and be engaged in an interaction. CADENCE currently models a single human agent and a single robot agent. The pronouns "you," "I," "my," etc. are parsed or generated based on speaker and addressee roles.

A *variable* is a feature of a specific object. Example surface forms are: "the block color," "the color of the block," "my blocks."

A *collection* is a group of semantic units of the same type. This is used for generating language of the form "x, y, and z."

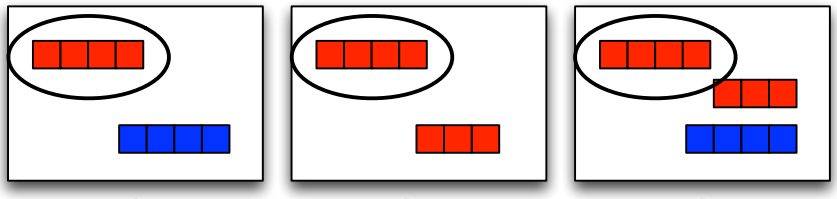
A *number* is self-explanatory.

8.3 Cognitive grammar

Next we define the natural language interface for composing and talking about these semantic primitives in a situated context. There are substantial bodies of work on parsing and generation that focus primarily on encapsulating syntactic grammaticality as evaluated by native speakers of a particular natural language. An alternative view that we subscribe to is promoted by the work of Langacker and other cognitive linguists, who argue that language understanding is a process of incremental conceptual construction [64]. Generally, it is pointless for a robot to have the ability to judge grammaticality of utterances without the ability to understand their meaning within its physical context.

CADENCE uses a custom bidirectional constituency grammar for natural language understanding and generation. Grammatical nodes are not only production rules for viable sequences of symbols in the language, but also incrementally construct the robot's semantic representation when parsed. Similarly, generation occurs by binding the semantic representation to a node, which gets decomposed into smaller constituents as they are passed down the hierarchy. Each node $N \in L$ includes:

- a set of grammatical function tags F in arbitrary languages (e.g. noun phrase or gesture stroke),
- a semantic type category T,
- a set of bound semantic entities E restricted by T,
- a set of actions A (refer back to Chapter 7 for how these are used),



Referring expression	Scene	Quality
the red block	1	optimal
	2	ambiguous
	3	ambiguous
the four by one block	1	ambiguous
	2	optimal
	3	suboptimal
the red four by one block	1	suboptimal
	2	suboptimal
	3	optimal

Figure 40: The ambiguity of referring expressions for the circled object changes based on the visual context, despite having identical attributes in each scene. "Ambiguous" means the object cannot be uniquely resolved, "suboptimal" means the object can be uniquely resolved but the expression is longer than necessary, and "optimal" means the object can be uniquely resolved and specifies a minimum number of features.

- an activation function $F_A(P, children(N)) \rightarrow E$ that operates on the child nodes of N to attach E (or instantiate conceptually grounded entities) to N,
- a generation function $F_G(E, parent(N)) \rightarrow A$ that decomposes information from the parent of N to bind a set E, then realizes the surface form for N and generates actions based on this form.

The activation and generation functions are always run on the specific situated context (more in Section 8.5). Figure 40 shows a simple example of how referring expressions can change depending on the visual context.

Previous work in situated language for robots focuses largely on the problem of learning associations between words and semantic representations from data [70, 33]. In this work we treat this associational process as domain authoring. One reason is that associational approaches are already predefining the semantic representations in

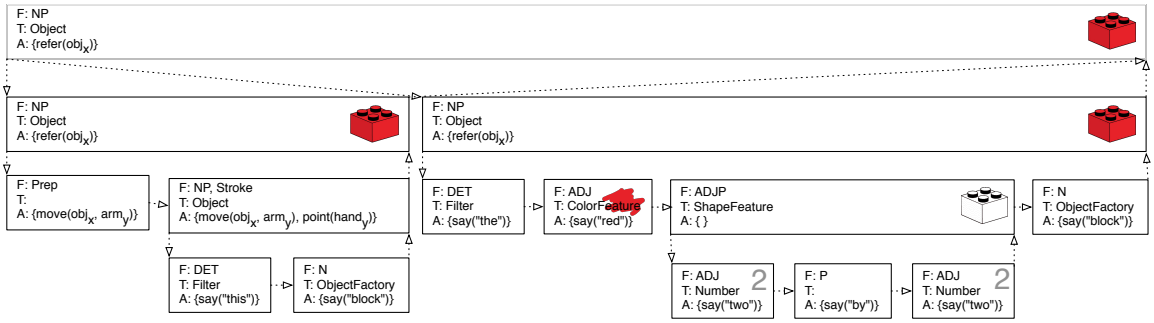


Figure 41: Example of cognitive grammar for an object noun phrase. The disjunctive root node represents two alternatives, one with speech only and one with speech + gesture.

the robot's perception or action space, which is most of the work, so it seems strange to pretend that the word associations are a mystery given that the semantics are designed in advance. The larger question of how to acquire core concepts in language usage (e.g. pronouns, modals, subjunctives) from lower-order cognitive substrates is not addressed by any of these previous approaches. Until it is, we might as well implement each rule once and be done with it.

Of course, there is a much more accessible view of language learning for robotics that focuses on acquiring new instances of semantic units using strong type assumptions. That is, there are only so many pronouns, but there are many more objects in the world, and probably a modest number of other actions that the robot can do. A logical extension to the semantic lexicon would be equip the robot with general-purpose object recognition and skill learning modules to define new entities for these two specific types during an interaction. With our current emphasis on collaboration within a closed domain, we leave such an extension to future work.

8.4 Records and propositions

At the topmost level, the root node of a tree for a dialogue act is bound to a *record*. A record is a mapping of semantic unit types to a set of semantic values of that type. We refer to the semantic type vector as the *semantic signature* of the record. A record

Type	Values	Surface form
Object	[window]	"the window"
Modal	[deontic$^+$]	"has to be"
Operator	[=]	""
FeatureValue	[blue]	"blue"

Table 8: Example of a complete record parsed from the statement, "The window has to be blue."

Type	Values	Surface form
Object	[door][roof][window][wall]	"it"
Modal	[epistemic$^+$]	"can be"
Operator	[>]	"more than"
FeatureValue	[3 wide][3 tall]	"three units"

Table 9: Example record for an ambiguous parse. The record has the same semantic signature as that in Table 8.

contains all of the information needed to parse out a *proposition*, which is a semantic statement that can have a truth value. Propositions may be mutable or immutable.

We say that a record is *complete* if every value set in the record has a cardinality of exactly 1. A complete record should parse to one proposition. Table 8 shows an example of a *complete* record. Note that the parser produced the implied semantic unit for "equals" even though there was no surface form explicitly attached.

Table 9 shows a record with the same semantic signature as the record in Table 8, but parsed from a sentence that was ambiguous in the context. Two of the entries can be resolved to multiple values, so multiple alternative propositions can be derived from these combinations. Rather than reasoning about all of these propositions, the dialogue system attempts to modify any incomplete record until it is complete. Section 8.7.8 for more details on these repairs.

Given a single semantic signature, there may be multiple syntactic forms that communicate the same semantic units. Take, for example, the following statements:

1. The roof and the door are the same height.

2. The roof and door height are the same.

These statements both have the same semantic signature of [Object$_1$, Object$_2$, Modal, Operator, Feature]. In these cases, the root node is disjunctive and has children representing the alternative syntactic forms.

Human conversation is often fragmented and contains non-sentential utterances, whether they are repairs, disfluencies, or collaborative completions. This record representation is designed to accumulate semantic information across multiple turns and can thus support more flexible turn-taking styles.

In addition, this representation enables a more precise definition of minimum necessary information:

> The minimum necessary information (MNI) of an act is reached at the earliest point that its record becomes complete.

That is, when all of the semantic units of the record can be uniquely resolved within the interaction context, then all of the information is available for the full interpretation of the act. With incremental processing, this can occur earlier than the end of an utterance or action.

8.5 Common ground

Common ground, as conceptualized by Clark and Brennan in [29], describes the body of knowledge, beliefs, and contextual assumptions shared between people who interact with each other. A complete model of common ground is enormously complex and would include models for cultural practices, lifelong memory, and the nuances of overhearing. Here we focus on the simplest essential slice for achieving mutual understanding with a human in the context of a single dyadic turn-taking interaction.

We define two *scopes* of knowledge, $Scope_{self}$ and $Scope_{common}$. $Scope_{self}$ describes a collection of propositions that the robot believes to be true, and that the robot has no reason to believe the human has knowledge about. $Scope_{common}$ describes a collection of propositions that the robot believes are shared between both the human

and the robot. Propositions can be added to or removed from the scopes as the interaction progresses, but the combined propositions from the two scopes always form a consistent set. Note that this model is simplified such that the robot does not explicitly model the human's beliefs; the robot would *not* pass the false belief test [126], cannot agree to disagree with a human, and does not account for deceptive behavior. $Scope_{common}$ is restricted to describing what the robot believes to be mutually accepted truths between both agents.

In addition, the robot maintains a *dialogue history*, which is a timestamped, ordered list of all collaboration acts that have occurred in the interaction. The robot's collaboration act processors examine the last N acts to decide what acts are appropriate to generate next, where N is small threshold set empirically.

The robot also has a *scene tracker*, which tracks objects that are known by both the robot and the human. This is required for accurate referring expression generation and resolution, which are contextually sensitive (see Figure 40). As a layer on top of frame-based perception, the scene tracker is responsible for monitoring consistent environment and object state over time, accounting for object permanence even in the presence of occlusions. Note that objects in the scene tracker do not necessarily need to physically exist in the agents' surroundings, but can also just be constructed as virtual entities to be referred to in a dialogue.

8.6 Collaboration acts

A *collaboration act* is a unit of intentional action in CADENCE. These are equivalent in scope to what is typically called a *dialogue act* in speech-only systems, but we expand the term in order to clarify that they need not include speech. For example, physical actions that progress a shared plan can also be collaboration acts. Collaboration acts are used by both agents to transfer information and cause side effects on common ground.

A collaboration act has a *speaker* and an *addressee*, which are agents participating in the interaction. Since CADENCE only models dyadic interactions, these agents are just the single human or the single robot.

The illocutionary *force* describes the underlying intention of the collaboration act, following speech act theory as defined by Austin [4] and later elaborated by Searle [102]. The force is used to determine contextually appropriate next responses in the interaction. In contrast with propositions, inappropriate responses are not deemed true or false, but rather felicitous or infelicitous. The distinction is of a semantic failure versus a pragmatic one.

The *topic* describes the primary semantic representation for the act's content. Sometimes it is a proposition, which can be accepted or disputed. Or sometimes it is a smaller semantic unit, like a feature or an object. The topic is backed by a *record* as defined in Section 8.4.

The *head binding* is the root node of the constituency tree following the grammar described in Section 8.3. If the speaker was the human, it is the parse tree produced from natural language understanding of the input text and common ground. If the speaker is the robot, the binding is used to generate actions to execute.

The collaboration act also holds pointers to the *previous* and the *next* act within the thread. Dialogue acts in conversation frequently follow the core structure of adjacency pairs, which describe two back-to-back turns from different speakers with complementary illocutionary forces [100]. These are known as the first pair part and second pair part. Some examples are greeting–greeting or question–answer. These responses can sometimes be linked together into longer chains. We call the entire chain a *thread*, even if the length is only 1 act so far. If the last item in the thread is a first pair part, we describe the thread as *open*, and if the last item is a second pair part, we describe it as *closed*. Thread states are used in arbitration and execution (see Section 8.8).

The collaboration act also has *side effects* that are applied when the act is committed to history. Side effects change the information state of the interaction and must be instantaneously executable. Specific examples of side effects will be described in the next section.

8.7 Processors

Next, we describe collaboration act *processors*, which are units of computation that interpret and generate collaboration acts. In this section we describe general-purpose act processors in CADENCE, which do not rely on any domain-specific content.

Conversational analysis also has the notions of *preferred* and *dispreferred* responses in adjacency pairs. Dispreferred responses are to be avoided in standard polite turn-taking, and are usually accompanied by distinctly different behavior such as hedges or apologetic attitudes. We also offer automatic strategies for toning down the effects of dispreferred second pair parts, such as rejections.

8.7.1 Inform

An inform act simply moves a proposition from self knowledge to common knowledge. That is, given a topic p, the act $Inform(p)$ has two side effects: $remove(p, Scope_{self})$ and $add(p, Scope_{common})$. These same side effects hold regardless of whether the speaker is the human or the robot. If the human communicates p when $p \in Scope_{common}$ already, the common ground state is unchanged.

A more conservative but also valid alternative is for the application of side effects to be delayed until the inform act is explicitly accepted by the addressee. The minor drawback in this formulation is that requiring repeated acceptances may be tiresome for the user, and when confirmations are not properly registered (speech recognition failure, subtle head nods, or no detectable response due to implicit acceptance), the robot will later attempt to repeat the information. Of course, more explicit user confirmation may also sometimes be desirable for the task.

The robot is motivated to increase common ground with the human as well as maintain engagement, so for all $p \in Scope_{self}$, the robot generates $Inform(p)$ acts over time whenever there is a lapse in the interaction. Stating new information out of the blue can be jarring to the human. To reduce the suddenness, these utterances are prefixed by a discourse marker such as "so," "well," or "by the way."

8.7.2 Accept/Reject

$Accept(p)$ or $Reject(p)$ are opposite possible responses to the truth value of p, but because they require the same computation, they are generated by a single $AcceptReject$ processor. When an open $Inform(p)$ has recently been introduced by the human in dialogue history, the $AcceptReject$ processor computes the consistency of p with all propositions in $Scope_{self}$, $Scope_{common}$, and the scene tracker. If p is consistent, the processor generates $Accept(p)$, and otherwise it generates $Reject(p)$.

Since $Accept(p)$ accepts p as being consistent, the side effects are equivalent to $Inform(p)$. The surface form is a backchannel like "okay" or "sure." $Accept(p)$ also automatically generates a head nod.

$Reject(p)$ has no side effects. The surface form for the illocutionary force is simply "no." It is also helpful to restate p with a reversal of polarity. For example, if the surface form of $Inform(p)$ was "The window can be blue," then the surface form of $Reject(p)$ would be, "No, the window can't be blue." $Reject(p)$ also automatically generates a gesture for shaking the head side to side.

A rejection is a dispreferred response, and the surface form of such a rejection still isn't very polite or helpful. The robot can improve upon such a response by offering a justification [106]. This discourse strategy requires the backend solver to have the ability to identify which propositions conflict with p. Given such a set C of conflicts, the justification is constructed from generating the collection of $Inform(c), \forall c \in C$. Thus, an example of the surface form for rejecting "The window can be blue" might

be, "No, the window can't be blue because the window and the door must not be the same color, and the door is blue."

8.7.3 Polar Query

Semantically, $PolarQuery(p)$ is actually identical to $Inform(p)$, but the illocutionary force is that of a question. The reasons for choosing one form over the other have to do with the amount of certainty that the speaker is communicating. Grammatically, the syntax of the act just reverses the subject-verb order compared to that of $Inform(p)$ and alters the prosody.

8.7.4 Feature Query

$FeatureQuery(v)$ is a request to know the value of a variable v; as a reminder, a variable is a feature associated with a specific object. A feature query uses the wh-form of the feature to ask a question like, "What color is it?"

8.7.5 Answer

The *Answer* processor answers both types of modeled questions, polar queries and feature queries.

As mentioned previously, $PolarQuery(p)$ is semantically equivalent to $Inform(p)$. To respond to a $PolarQuery(p)$, the *Answer* processor performs the same computation as *AcceptReject* in order to determine the consistency of p with previous knowledge. If p is consistent, the processor generates $Yes(p)$, which differs only from $Accept(p)$ by using "yes" instead of "okay" as the surface form. Otherwise it generates $Reject(p)$.

To answer a $FeatureQuery(v)$, where v represents the feature f in object o, the processor looks up the feature value for f in o. If there is a value associated with f, the processor generates $Answer(v)$. An example of an answer in response to the question, "What color is the window?" would be, "Blue," or more completely,"The

window is blue." If no value is returned from the lookup, the robot says, "I don't know."

When answering a propositional question (that does not implicitly carry positive or negative connotations), having any answer is preferred regardless of the polarity of the answer, and saying "I don't know" is dispreferred. In an effort to be more helpful, the robot can follow up the response with semantically relevant propositions. A set of relevant propositions R from $Scope_{self}$ and $Scope_{common}$ can be enumerated and ranked based on whether their record values contain v, f, or o. Just like when constructing a justification, the processor can generate and concatenate $Inform(r), \forall r \in R$. The resulting response is of the surface form, "I don't know, but the window and the door can not be different colors."

8.7.6 Proposal

$Propose(p)$ is an act that can be generated when $p \notin Scope_{self}$ and $p \notin Scope_{common}$. This requires that the task solver have an ability to compute new propositions in an effort to progress the task. The surface form of $Propose(p)$ is similar to $Inform(p)$. An example of a proposal surface form is, "Let's make the door red" (in contrast to "The door must be red" for $Inform$). The semantics are also essentially equivalent to $Inform(p)$; the difference is mostly in the preconditions and the computation.

There are several ways to model the side effects of contributing a proposal to common ground. One way is to treat it identically to $Inform(p)$, in which p is added to $Scope_{self}$ after the act is executed (note, $remove(p, Scope_{self})$ does nothing because a precondition for a proposal is $p \notin Scope_{self}$). A more conservative alternative is to delay this side effect until the addressee has explicitly accepted the proposal at the next $Accept(p)$. In the experiment described in the next chapter, we used the former approach, and observed a split in how participants interpreted this act. Some appeared to assume that information as truth going forward without verbally

accepting it, while others appeared to ignore information proposals that they did not explicitly verbally accept. So both options are understandable, but there is room for further improvement.

For a higher-initiative robot, a proposal can also be given as an answer to a feature query. For example, in answering the question, "What color should the door be?" a low-initiative response would be to say "I don't know" when the lookup returned nothing. A higher-initiative response would be to compute a consistent possible value and respond, for example, "The door could be red" or "Let's make the door red."

8.7.7 Request

Request(a) is a request for the addressee to perform action a. The surface form can be a command, i.e. "Pick up the red block," or a question, "Can you pick up the red block?"

8.7.8 Repair

The *Repair* processor operates in response to acts that have an incomplete record. The goal of acts generated by the *Repair* processor is to resolve ambiguities or fill empty slots in those records so as to complete them. The resulting complete record can then be parsed normally into another collaboration act. The acts generated by the *Repair* processor all fall into a category called *clarification requests*. Ginzburg gives an in-depth overview of clarification requests in natural human conversation in [39]. Bohus also implements many examples of repairs in[11].

The simplest type of clarification request is to ask for a repetition. We define an action *Repeat*, which is to repeat the last spoken act of that speaker. *Request*(*Repeat*) thus gives the question, "Can you repeat that?" The human or robot can also simply ask "What?" as an alias for *Request*(*Repeat*). *Repeat*(*Request*) serves as a fallback for when other types of repairs are not applicable. Here we also add that it is inadvisable for the previous speech act selected by *Repeat* to also be a *Request*(*Repeat*),

lest a loop ensue (a mistake we uncovered in the experiment described in the next chapter).

Since each semantic unit type provides a wh-form, another repair strategy is to perform wh-substitution for all entries whose cardinality is not equal to 1. For example, Table 9 displays a parse where two of the semantic units have multiple possible values. An example of this type of repair is to respond with, "What can be more than what?" When there are multiple wh-replacements, this repair usually leads to a repetition or paraphrase of the original propositional content by the addressee. When there is a single wh-replacement, a fragment representing that exact semantic unit is a more common response. The robot thus also has a *Fragment* processor that can parse a non-sentential phrase and merge it into the appropriate entry in a previous record according to its semantic type.

There are other repair acts possible when repairing a single semantic unit in a record. Let's assume that the human has stated, "The block has to go here," but no location was resolvable to "here" due to errors in pointing detection. One possible repair act is a *reprise sluice*, in which only the wh-form of the ambiguous unit is used: "Where?" If there are several possible options for the ambiguous value, robot could also propose one: "Here?" (with a deictic gesture), and wait for confirmation or correction. While we have implemented these repair forms, we have not characterized the exact situations in which one form should be generated over another.

8.8 Timing considerations

In Chapter 6 on floor regulation, the turn-taking model we created was able to seize the floor based on various timing parameters. The model was demonstrated in a semantics-free domain in which the robot used an artificial language. With the ability to have deep semantic dialogues as described in this chapter, the robot must apply timing rules according to discourse structure. We briefly highlight how timing within

interactions is influenced by the dialogue representations described so far.

8.8.1 Regulators

To regulator the timing of turns, each collaboration act processor is installed in the dialogue manager with a corresponding *regulator*. A regulator uses a resource monitor (described in detail Section 7.1) to provide a timing rule for when a processor can generate collaboration acts.

Previously, we described the importance of adjacency pairs to the sequencing of collaboration acts in interaction. One timing rule is that the second pair part has a tight temporal constraint relative to the first pair part of an adjacency pair. This is the *response delay*. When there are multiple competing acts that can be executed, second pair parts take priority. It is also possible for response delays to be increased for dispreferred second pair parts. Schegloff explains in [99] that this delay can provide an opportunity for the first part speaker to restate the original content in a way that saves the addressee from the discomfort of speaking a dispreferred response.

In a conversation, both agents are motivated to continue contributing in a timely fashion to one thread before opening a new one. This is because of reduced ambiguity and increased brevity due to the ability to make more contextual assumptions. For example, if an agent says "Okay" while many topics are being discussed, it can be ambiguous which proposition the agent is actually accepting. Thus, we have the robot only start new threads (e.g. "By the way, the window has to be...") after a *lapse* in the interaction, which is an extended period when neither participant speaks. This ensures that both parties have enough time to contribute to existing threads first.

Increasing delays before starting new threads also prevents the robot from taking too much control over the dialogue, when the desired dynamic is a balanced collaboration. Analysis of expert-client dialogues by Whittaker and Stenton showed that control tends to stay with one speaker over multi-turn phases [123]. In previous work,

we found that allowing the robot to maintain total discourse control by having no delays between thread starters had detrimental effects on human mental models [20]. Such discourse control can be partially regulated through the timing of delays and turn durations.

We define four regulators, from shortest to longest delay:

1. **Resource free** – act as soon as the resource becomes available.

2. **Respond** – wait a short response delay (less than one second) to continue the thread.

3. **Initiate** – initiate a new thread. Wait slightly longer to allow the interaction partner to continue the current thread if desired (more than one second).

4. **Avoid lapse** – act only after an extended lapse, in an effort to continue the interaction (at least several seconds).

8.8.2 Incremental side effects

In addition, now that we have word-level semantics, we can have common ground state change incrementally over time rather than only at the end of a turn. This results in more correct state changes when the robot is interrupted. Let's take the example of the robot's turn being: "No, the window can't be blue because the window and the door must not be the same color, and the door is blue." The act is committed when the minimum necessary information has been communicated, which we define to be the illocutionary force plus the minimal semantic content of the act. In this case, the MNI is passed after "no" is uttered (or more conservatively, after the proposition is restated in "the window can't be blue"). Then for each $Inform(p)$ in the justification, p is moved to $Scope_{common}$ as soon as the last word in the inform act is uttered. If the robot is interrupted midway, the common ground state should still accurately reflect what has been partially stated.

8.9 Processing pipeline

Figure 39 gives an overview of how the components described in this chapter interact with each other. The entire system updates at approximately 30 Hz.

Whenever a human takes speaking turns, voice activity detection (VAD) produces the low-level signal for spoken turn-taking. Automatic speech recognition (ASR) produces strings that are parsed into possible semantic representations using NLU. These mappings of text to semantic forms are called *interpretations*. If the human takes multiple consecutive turns, all of these are accumulated in a buffer.

When the time is appropriate for the robot to take a speaking turn according to the speech turn-taking process, the dialogue manager arbitrates among the possible interpretations by selecting the combination of non-conflicting interpretations that covers the maximal number of words in the input text. (Conflicting interpretations are those assigned to the same words.) For equally-scored combinations, the arbiter breaks ties based on the completeness of parsed records. The winning combination of interpretations is committed to dialogue history.

Then, the dialogue manager runs all of its collaboration act processors based on the common ground state. The result is a set of a collaboration acts that would be semantically and contextually appropriate to take next. These collaboration acts are generated and realized with the assistance of the NLG module. We refer to this possible set of acts as *options*. Processors are also responsible for continually revoking acts that are no longer relevant.

The execution of options can be somewhat flexible. One approach is to use a scheduler to determine how to queue all of the actions from the options into the timed Petri net based on their resources required, as described in Section 7.4. Another is to have an option arbiter that selects a minimal set of acts that can be executed simultaneously – functionally a shorter-horizon version of scheduling. For a small but highly dynamic domain, a task-specific arbiter may be more appropriate. Either way,

the result is a process that queues action primitives into the Petri net at the right time to start execution without risk of deadlock.

During action execution, the robot can hesitate or interrupt its actions based on resource availability, which is modeled on top of continuously monitoring the human's behavior. For example, the robot interrupts speaking when the human speaks or yields its arms based on the human's movements. As actions are completed, they are committed to dialogue history and apply side effects to common ground.

CHAPTER IX

EVALUATING COLLABORATIVE DIALOGUE

In our last experiment, presented in Chapter 6, we investigated multimodal turn-taking with speech, gaze, gesture, and manipulation in an open-ended domain absent of semantic understanding. We now proceed to evaluate turn-taking in a domain where the human and the robot must collaborate to reach common ground in a task that involves both dialogue and manipulation. Here, mutual understanding and agreement is critically important to task success. In this semantically rich interaction setting, we rely on the advances in multimodal resource modeling from Chapter 7 and situated dialogue from Chapter 8 to round out CADENCE as a framework for joint cognition and action.

The multimodal resource model developed in Chapter 7 was motivated by shortcomings of a unitary resource assumption in modeling turn-taking. A consequence of the updated model is an ability to execute intentions concurrently, when the resource state allows it. In this chapter, our primary goal is to investigate the effects of that concurrency on turn-taking dynamics and task outcomes. Our experiment is therefore designed to compare the resource model from Chapter 7 against a baseline of single-intention execution, in a setting of situated collaboration enabled by the implementation of Chapter 8.

A secondary interest in the design of this domain is the question of repairing misunderstandings is interaction. A thematic challenge of robotics is acting in dynamic environments with sensor noise. These challenges are compounded in interactions with humans, where the human mental model is not directly observable, and in multimodal perception and action, where disagreement across modalities creates more

ambiguity. At the same time, we challenge that the multiple modalities can also be leveraged to recover from these errors when both participants are motivated to reach common ground. In this chapter, we aim to show that through interaction, a robot and a human can converge on a shared mental model even in spite of error-riddled vision, speech recognition, and manipulation.

First, we describe the design and implementation of a dyadic interaction scenario where the human and the robot share a goal to construct a model together, but each has partial information about the model's characteristics. They thus must have a dialogue in order to gain all of the necessary information, and they also must physically build the model. The robot's dialogue system is equipped with several strategies for maintaining common ground. We then conduct an experiment with two conditions in which the robot exhibits a different turn-taking style in each condition. In the baseline condition, the robot executes intentions sequentially, and in the experimental condition, the robot can execute intentions concurrently based on resource availability modeled by CADENCE. We then characterize differences in performance and behavior that result from these contrasting turn-taking styles.

9.1 Task description

The human and the robot are tasked with designing and building a model together using large colored blocks (Mega Bloks). The model has to satisfy a total of 10 design requirements, but each agent starts the interaction only knowing 5 requirements. These starting sets are mutually exclusive. Even when combining all 10 requirements, the model is underconstrained, so the agents must also make some decisions about the model's remaining characteristics.

As can be seen in Table 10, the starting constraints were designed deliberately with dependencies across the agents. For example, the robot has a constraint on the window width, and the human has a relative constraint between the door and the

Table 10: Human and robot task constraints

Robot constraints
The door has to be at least 3 units tall.
The door and the roof cannot be the same color.
The house has to be below row 2.
The wall width has to be greater than the wall height.
The window has to be at least two units wide.
Human constraints
The window has to be blue.
The roof height has to be greater than the door height.
The window has to be left of the door.
The house has to be to the right of column C.
The door and the window have to be the same width.

window width. This design was intended to encourage more collaborative problem-solving, rather than having the agents act relatively independently.

9.1.1 Semantics

As described in the previous chapter, the semantic lexicon in CADENCE is designed to easily install domain-specific knowledge modules that can plug in to a more general-purpose dialogue and reasoning system. Here we review the specific semantic components for this task and how they are represented to operate within a constraint optimization framework that integrates language, vision, and spatial reasoning.

For simplicity, we have designed this task to focus on two-dimensional imagery in a discrete grid with a fixed frame of reference. Although CADENCE supports more sophisticated language and spatial reasoning than described here, we simplified the task design in order to reduce other engineering and logistical problems in the task, such as implementing perception and language for detecting and repairing occluded parts of the model. It is a relatively straightforward exercise to consider generalizations of the described framework.

For this task setting, all variables were either discrete or integer-valued. Thus, the

task could be solved within an integer linear programming framework [44]. The base set of domain constraints are shown in Table 11 and Table 12. Next, we elaborate on the components of the task.

9.1.1.1 Objects

The task objective was to build a *house*, which consisted of a *roof* over a *wall*, and a *door* and a *window* inside of that wall. These last four objects formed a subcategory of house *parts*. All parts were rectangular, except for the roof whose shape decreased in width from the bottom to the top edge. Size and shape rules about the parts of the house are shown in Table 11.

The task involved building a house out of blocks on a green building plate. The plate had 11 *rows* and 12 *columns*, which were referred to using natural language as rows 1–11 and columns A–K to reduce confusion. A row was defined as a spatial region on the plate that was one unit high and the same width as the plate; a column was a region that was one unit wide and the same height as the plate. The feature values of rows and columns were immutable.

Another object type was a *segment*, which was a connected component of a single color that had been built from blocks on the building plate.

9.1.1.2 Features and feature values

All of the objects had integer features for *width* and *height*, and a location represented by four integer coordinates: the *left* edge, the *right* edge, the *top* edge, and the *bottom* edge. Although four features could have represented this information rather than six, the additional features were created for convenience of referring.

All of the objects except rows and columns had a discrete *color* feature, for which the values could be red, yellow, or blue.

The *segments* additionally had a discrete feature of *part*. That is, a segment assigned the part of "roof" was thought to be a partially constructed fragment of the

roof. Multiple segments could be assigned to a part. A segment assigned to a part had to match that part in color and be inside of that part's bounding coordinates.

9.1.1.3 Operators

Operators are used to constrain feature values, and map relatively directly to the underlying constraint representation.

For numerical features, the operators were *greater than* (>), *less than* (<), *the same as* (=), *different from* (! =), *at least* (>=), and *at most* (<=). Operators could be stated relative to a numerical value or to another feature value, i.e. "the door is at least three units tall" or "the roof height is greater than the door height." For the features of width and height, superlatives were also modeled ("taller than," "wider than").

For discrete features, feature values could only be equal or not equal. Discrete features values were defined relative to a feature vector of binary indicator variables summing to 1; the indicator was 1 for the index of the assigned feature value. Operators relative to a feature value ("the house color is red") imposed a constraint on the indicator variable at that feature value's index. For operators relating two features ("the window and the door are the same color"), constraints were added between all indicator variables of the same index (see top of Table 11).

9.1.1.4 Spatial relations

Spatial relations impose groups of constraints on the coordinates of multiple objects. Table 12 shows the relations that were relevant to defining this task. Although all of these were needed in the task solver, only *Left*, *Right*, *Above*, and *Below* were exposed in the natural language interface during the interaction.

Spatial relations are not intrinsically different from operators, except that they are parameterized on objects instead of directly on features. For example, saying that the bottom coordinate of x has to be higher than the top coordinate of y is equivalent

to the optimizer to saying that x is above y. The object features are simply implied in the spatial relation.

9.1.1.5 Summary

In summary, the semantic lexicon for the task was:

- Objects – house, door, roof, window, wall, segment, rows, columns
- Features – color, width, height, left, right, top, bottom, part
- Operators – greater than, less than, the same as, at least, at most, different from
- Relations – left of, right of, below, above

9.1.2 Model construction

The task pipeline is shown in Figure 42. The task solver combines all of the basic domain constraints, the constraints in the robot's self-knowledge scope (i.e. the robot's 5 requirements), and the constraints in the common ground scope into a single optimization problem. The objective function here was to minimize the size of the house ($w_{HOUSE} + h_{HOUSE}$). A feasible solution contains feature values for all of the objects' features in the task, such as the roof is red, the house is 5 units wide, etc. The robot's mental model of the house is then generated by filling an occupancy grid based on these feature settings. This mental occupancy grid represents the goal state that the robot intends to achieve physically.

To generate actions in the real world, the robot has to palletize the goal occupancy grid based on the blocks in the available workspace. For each available block in the workspace, identified by shape and color, all feasible placements (location and orientation) are enumerated that would satisfy the goal state. Each pair of placements also has a binary variable indicating whether the placement is allowed to co-occur,

Table 11: Domain definitions

Description	Notation
Discrete feature	$values = \{v_0, v_1 \dots v_N\}$
	$\sum_{n=0}^{N} i_n v_n = 1,\ i_n \in \{0, 1\}$
Discrete $v = w$	$\forall n, i_{n_v} - i_{n_w} = 0$
Discrete $v \neq w$	$\forall n, i_{n_v} + i_{n_w} \leq 1$
Width, height, color of object x	$w_x,\ h_x,\ c_x$
Coordinates of object x:	$t_x,\ b_x,\ l_x,\ r_x$
top, bottom, left, right	$b_x = t_x + h_x - 1$
	$r_x = l_x + w_x - 1$
House dimensions	$h_{HOUSE} = h_{ROOF} + h_{WALL}$
	$t_{HOUSE} = t_{ROOF}$
	$l_{HOUSE} = l_{ROOF}$
	$r_{HOUSE} = r_{ROOF}$
	$b_{HOUSE} = b_{WALL}$
The roof covers the wall	$Covers(ROOF, WALL)$
The door touches the bottom edge	$b_{DOOR} = b_{HOUSE}$
The window can't touch the bottom	$b_{WINDOW} < b_{HOUSE}$
The window is inside the wall	$Inside(WINDOW, WALL, 0)$
The door is inside the wall	$Inside(DOOR, WALL, 0)$
Adjacent parts can't be the same color	$c_{WALL} \neq c_{ROOF}$
	$c_{WALL} \neq c_{DOOR}$
	$c_{WALL} \neq c_{WINDOW}$
House color is defined to be the wall color	$c_{HOUSE} = c_{WALL}$
Certain parts can't overlap	$Outside(WINDOW, ROOF, 1)$
	$Outside(DOOR, ROOF, 1)$
	$Outside(WINDOW, DOOR, 1)$
Segment s is grounded to part x	$IfThenEqual(i_x, c_s, c_x)$
	$IfThen(i_x, Inside(s, x))$
Roof shape	$h_{ROOF} - 0.5 w_{ROOF} \leq 0.5$

Table 12: Semantic relations

Description	Notation
x covers y	$Covers(x, y)$: $l_x \leq l_y$, $r_x \geq r_y$, $t_y = b_x + 1$
x is to the right of y	$Right(x, y)$: $l_x > r_y$
x is to the left of y	$Left(x, y)$: $r_x < l_y$
x is above y	$Above(x, y)$: $b_x < t_y$
x is below y	$Below(x, y)$: $t_x > b_y$
x is inside y (m from y's edge)	$Inside(x, y, m)$, where $m \in \mathbb{Z}$: $l_x + l_y \geq m$, $r_x + r_y \leq m$, $t_x + t_y \geq m$, $b_x + b_y \leq m$
x and y don't overlap (separated by m)	$Outside(x, y, m)$, where $m \in \mathbb{Z}$, and $M \in \mathbb{Z}$ is large: $\sum_n i_n \geq 1$, $i_n \in \{0, 1\}$, $l_x - r_y - Mi_{right} \geq m - M$, $l_y - r_x - Mi_{left} \geq m - M$, $t_y - b_x - Mi_{above} \geq m - M$, $t_x - b_y - Mi_{below} \geq m - M$
if condition c, then proposition p	$IfThen(c, p)$: given $p \rightarrow w_1x_1 + w_2x_2 \ldots \Diamond m$, and $M \in \mathbb{Z}$ is large: $w_1x_1 + w_2x_2 \cdots - Mc \Diamond m - M$
if condition c, then discrete $v = w$	$IfThenEqual(c, v, w)$, where $c \in \{0, 1\}$: $\forall n, i_{n_v} - i_{n_w} - Mc \geq -M$, $\forall n, i_{n_v} - i_{n_w} + Mc \leq M$

meaning that they do not intersect. The palletizer then solves for a valid combination of placements that minimizes the total number of placements and the number of leftover cells in the goal state. These leftover cells can be filled with 1x1 blocks by the human.

Given the palletized goal state, the robot generates pick and place actions for the workspace blocks based on kinematic feasibility. The robot's manipulation planning also has certain heuristics such as preferring larger and closer blocks.

Perception of the task state and workspace was performed using an overhead Asus Xtion pointing straight down at the table (Figure 43). The table was subtracted using a plane estimation approach, leaving behind clusters of points representing non-planar objects in the scene [115]. Any cluster attached to the edge of the frame was considered a robot or human arm and filtered away. The cluster corresponding to the building plate was binned into an occupancy grid based on prior knowledge of the building plate dimensions. Other clusters represented blocks on the table and were classified into pre-trained shape and color categories. The blocks' location, orientation, and state of occlusion were also estimated and tracked to be used for manipulation.

Since the goal state changes dynamically throughout the interaction based on new information, the physical building plate can end up being occupied by blocks that are not needed. The robot thus continually monitors the differences between the physical state and the goal state so that it can attempt to repair regions in the physical state that it considers mismatched.

9.2 Dialogue

The previous chapter described the functionality of several general-purpose dialogue acts implemented in CADENCE. Here we briefly review the acts used specifically in this task domain. As a reminder, dialogue acts are continuously (i.e. at approximately

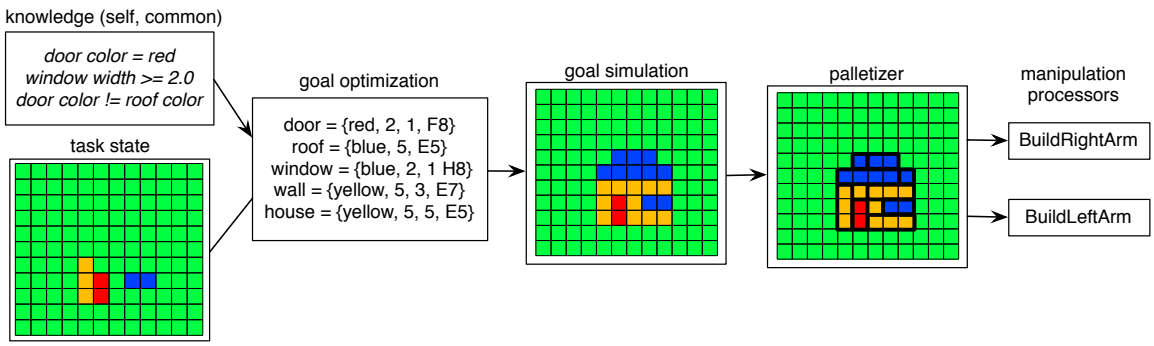


Figure 42: Task pipeline for house building collaboration. The goal state is optimized based on constraints in the robot's knowledge, as priors or introduced through dialogue, and the current state in the world. The palletizer generates block placements based on available blocks, which are used to generate pick and place actions.

30 Hz) interpreted from the human and generated (or revoked) as options by dialogue act processors.

9.2.1 Acts

Both the human and the robot can use the *Inform(p)* dialogue act, where proposition p is one of the task requirements. An example is the statement, "The window has to be blue." To avoid overloading the human with too much information too quickly, the robot may only offer such an inform after a lapse in the interaction.

Both the human and the robot can use the *AcceptReject(p)* dialogue act, to accept or reject a proposition p. The robot generates such acts whenever the human introduces new propositions.

The human can ask a *PolarQuery(p)*, of the form, "Can the window be blue?" as well as a *FeatureQuery(f)*, of the form, "What color should the window be?" The robot is not able to ask these types of queries about the task parts.

9.2.2 Common ground maintenance strategies

Because errors and ambiguities are common occurrences in interaction, especially when agents have partial information and imperfect sensing, it is essential for agents to be able to repair misunderstandings. The robot's task dialogue uses the following

strategies to maintain common ground.

9.2.2.1 Justification

As mentioned in the previous chapter, justification can be used when rejecting a proposition in order to increase transparency. Accordingly, the robot employs justification in this task. If the additional constraints implied by a proposition result in the task optimizer returning an infeasible solution, the irreducible infeasible set (IIS) of constraints is computed. These constraints are mapped back to their high-level semantic form, and after removing the initial proposition p, are enumerated as a collection of propositions. A justification following a rejection takes the form: "No, the roof can't be red because the door and the roof can't be the same color, and the door is red."

9.2.2.2 Propositional repairs

Because of errors in speech recognition, incorrect propositions can be introduced into common ground. For example, the human says "the door must be red," and the speech recognizer provides "the roof must be red." If the proposition is consistent with the robot's knowledge so far, it is added to common ground, introducing a false constraint. Later, this proposition must be repairable. There are many discourse strategies possible for conducting this repair, but here we implement just one method.

The *AcceptReject* processor handles these cases as follows. Let's assume that a false proposition p_F was initially introduced into $Scope_{common}$ due to an ASR error. When a new proposition p' is proposed by the human that conflicts with p_F, *AcceptReject* computes the IIS conflict set C. If $|C| = 1$ after removing p_F from C, the robot accepts the new p' and removes p_F from $Scope_{common}$. If $|C| > 1$, then the robot's justification provides transparency about the conflict set C. The human can then choose to repair individual propositions in C.

The exception is that propositions in the robot's knowledge that are set as immutable will *not* be replaced by new conflicting propositions. In these instances, the robot will adamantly reject the new information and offer the immutable propositions in the justification.

9.2.2.3 Clarification requests

The robot also uses clarification requests to communicate speech misunderstandings. As described in the previous chapter, wh-substitution can be used to produce more specific repairs. For example, the robot can ask, "The door has to be what?" to solicit a missing feature value from a proposition. For this experiment, the threshold for wh-substitution was set to 1 unit – that is, when more than 1 semantic unit is misunderstood, the dialogue act falls back to a repeat request ("What?"). The human can also request repeats from the robot.

9.2.2.4 Grounding queries

When the house is only partially constructed on the plate, there is often ambiguity as to which parts the segments are supposed to be assigned to. Thus, the robot is also able to ask the polar question, "Is this x?" where x is a referring expression for one of the task parts. This is essentially a $PolarQuery(p)$, except the robot only asks polar queries about this specific feature rather than any object-feature pair. Segments do not have a speech-only surface form, and are referred to as "this" with a simultaneous deictic gesture.

Technically, this act was available to the human as well, but perception of human finger pointing was not accurate enough for this to be interpreted reliably. Participants were warned of this in advance.

Processor	Regulator	Example surface forms
Greet	Respond	"Hello."
CompleteTask	Initiate	"I think we're done."
AcceptReject	Respond	"Okay."
		"No, the door can't be two units tall because it has to be at least three units tall."
PolarQuery	Initiate	"Can it be blue?"
FeatureQuery	Initiate	"What color is it?"
		"How wide should it be?"
AnswerQuestion	Respond	(Feature) "The door is red."
		(Polar) "Yes, the door is red."
		(Polar) "No, the door has to be yellow."
Fragment	Respond	"The roof."
		"Red."
InformConstraint	Avoid lapse	"By the way, the window has to be blue."
GroundScene	Initiate	"Is this the door?"
BuildRightArm	Regions free	$pick(block, table)$
		$place(block, table)$
		$place(block, mat)$
BuildLeftArm	Regions free	(same as right arm)
RequestRemove	Initiate	"Can you remove this?"
Repeat	Respond	(repeat the last speech act)
Reprise	Respond	"The door has to be what?"
		"What?"

Table 13: The collaboration act processors used in the experiment, in order of priority. Both conditions use the same set. *Regulators* control when processors are able to generate new acts, and are defined in Section 8.8.1 on 148.

9.2.2.5 Requests for action

The robot does not have the manipulation dexterity or strength to remove blocks from the mat. When there are sections on the building plate that the robot considers to be conflicting with the goal state, the robot can request that the human remove the section with the directive, "Can you remove this?"

9.3 Experiment

The experiment was intended to evaluate the effects of manipulating the robot's multimodal concurrency when collaborating with a human. The experiment was

between-groups with two conditions, which we call the *sequential condition* (baseline) and the *concurrent condition* (experimental). The conditions differ in how they execute actions when the robot has multiple intentions at a time. We hypothesize that the experimental condition, which uses CADENCE for concurrent execution, will produce more fluent behavior and therefore more successful task performance.

9.3.1 Conditions

9.3.1.1 Sequential condition (baseline)

In this condition, the robot executes a single atomic intention at a time. This condition characterizes the behavior of state-based systems in which each intention is a single atomic state and only one state is active at a time. Such a formulation is much simpler than modeling resources or concurrency; if only one intention is executed at a time and it is never interrupted, there is no concern about resource contention across multiple intentions.

9.3.1.2 Concurrent condition (experimental)

In this condition, the robot adheres to the multimodal resource model described in Chapter 7. The robot can execute multiple intentions at a time if the resources are available to do so, and also interrupts actions based on social awareness of these resources. Given that the baseline is a simpler approach, the goal of the experiment is to determine the value of performing additional reasoning about resources. We hypothesize that additional resource awareness will make the robot more responsive, improving the fluency of the dyad.

9.3.1.3 Environment setup

Figure 44 depicts the environment setup for the task. The human and the robot faced each other from across a long table. Centered on the table was a green 12x11 building plate, on top of which the house model was to be constructed. The human

was allowed to use any of the blocks, but the robot could only perceive and reach blocks in a limited area, as demarcated in the figure. The human therefore controlled the blocks accessible to the robot by moving them into the robot's workspace. The human was also instructed to keep the blocks separated from each other to facilitate the robot's manipulation.

Two sheets of paper were taped to the table for the human to refer to. One was a reference sheet for what utterances were within grammar for the task. The other was a list of the human's 5 requirements.

A overhead Asus Xtion was located over the building plate and was used for the robot's perception of the task progress and available blocks. A Kinect was mounted above and behind the robot's head in order to track the human's head and hand locations. A video camcorder was stationed at the end of the table on the human's right side, to be used for video coding.

The human wore a headset for speech recognition. Speech recognition was performed using a grammar with InproTK [9], an open-source incremental processing toolkit built on top of CMU Sphinx [63]. InproTK returned voice activity, incremental ASR results, and final ASR results for the user's speech model.

9.3.2 Population

Participants were recruited from campus mailing lists and thus comprised students, staff, and friends of students or staff. The recruitment email did not advertise any compensation, although several participants were incentivized by receiving extra credit in an AI class for their participation. During the explanation phase for the task, participants learned that they would be eligible to receive a $50 Amazon gift certificate if they completed the task entirely correctly.

The first 9 participants were treated as a pilot study, during which the explanation phase was modified until the task could be understood well enough to be completed

successfully. After the pilot study, participants were assigned in alternating order to the two conditions while maintaining gender balance. Interactions with 5 participants had to be terminated prematurely due to some part of the system crashing and thus were omitted from analysis. The final data set represented a total of 26 participants, with 13 participants in each condition (8 male, 5 female). Participants were 18–45 years old, with a mean age of 23.6.

9.3.3 Protocol

After signing a consent form, the participant was instructed to read a 3-page primer explaining the task and interaction. The complete document is given in the appendix. The primer described the task semantics, the human's 5 design requirements, and limitations of the robot in the interaction. After reading about the task, the experimenter answered any initial questions about the document. The participant then practiced speaking examples from the grammar into the speech recognizer with the guidance of the experimenter. Finally, the participant practiced the interaction briefly with the experimenter, with corrections from the experimenter if the participant spoke off

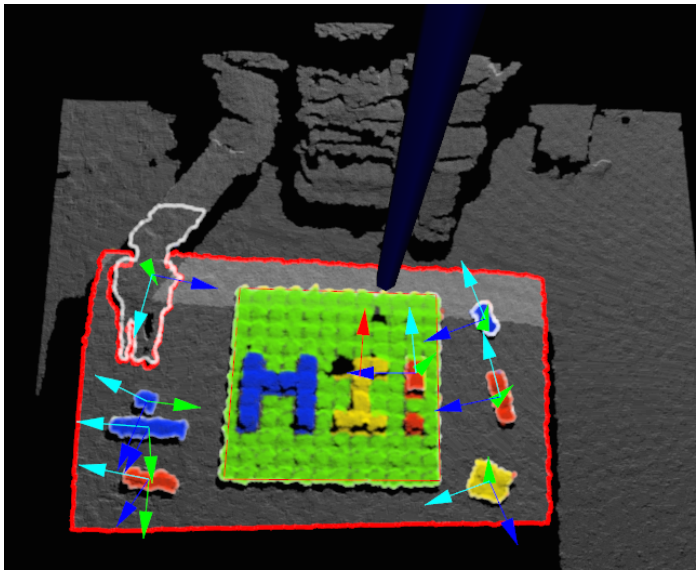
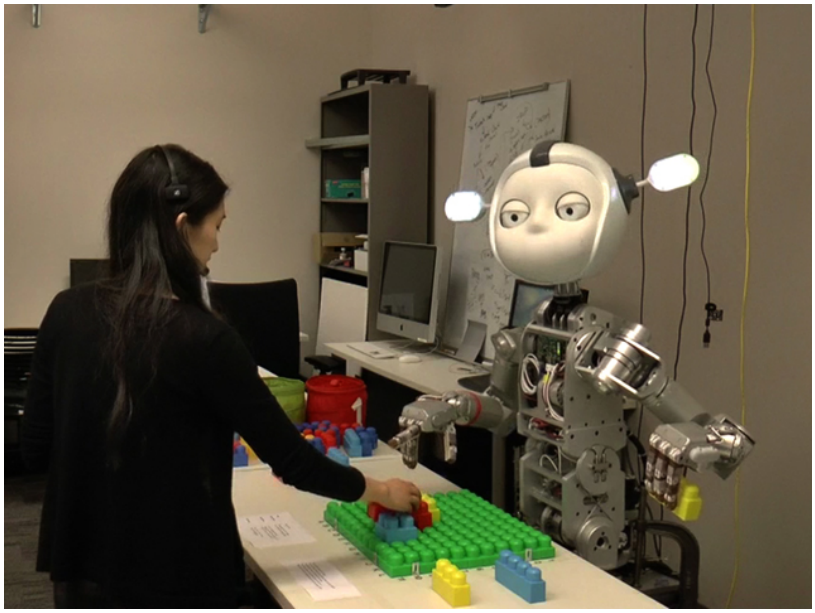
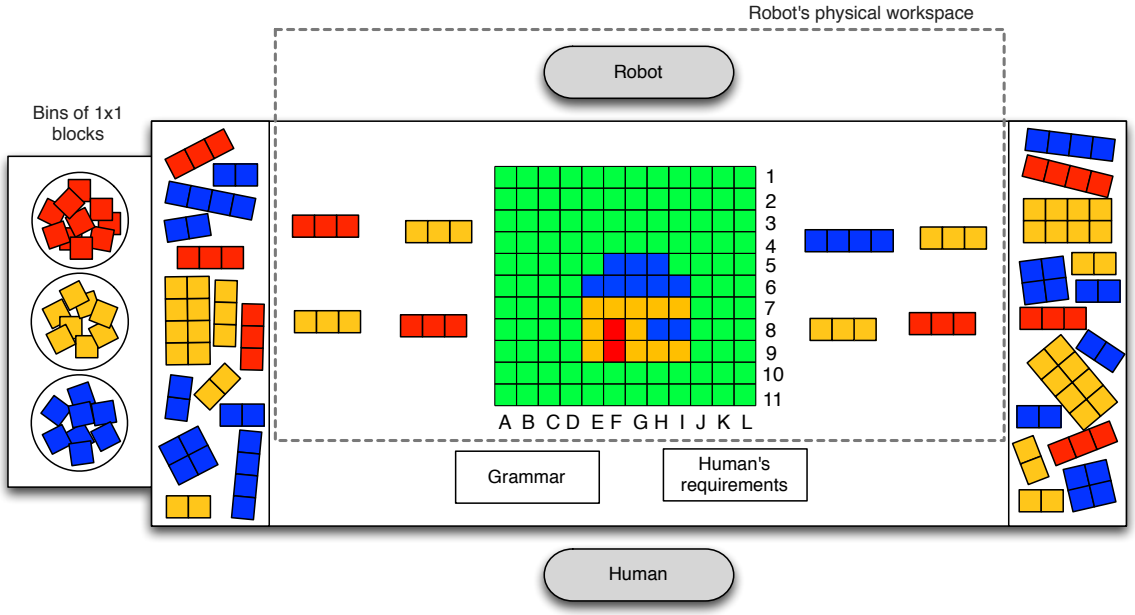


Figure 43: The state of the building plate and the blocks in the robot's workspace were tracked with point cloud perception using an overhead rgbd sensor.



(a) A human collaborating with the robot



(b) Overhead diagram of workspace

Figure 44: The robot and human faced each other from across a table, building a house oriented towards the human. The human could use any blocks on the table and controlled the blocks accessible to the robot.

grammar.

The participant was given a maximum of 15 minutes to complete the task, but could end sooner if desired. The human was instructed to start by greeting the robot, and the start time was taken to be the timestamp when the robot finished greeting the human. The participant could refer to the time passed during the interaction on a nearby iPad. The interaction was terminated either by the participant pressing a button on the table or when 15 minutes were up. During the interaction, participants had access to "cheat sheets" taped directly in front of them on the table for the speech grammar and their 5 design requirements.

After interacting with the robot, participants filled out a survey about their subjective experiences.

9.3.4 Survey

1. How did you find the pacing of the interaction? (1 = very slow, 7 = very fast)
2. Who led the interaction? (Simon, me, about equal)
3. On a scale from 1–100, how much did you contribute toward the task solution? (50 means both contributed equally)
4. Did you complete the task successfully? (yes, no, I don't know)
5. Please rate the following statements about the interaction with Simon. (1 = strongly disagree, 7 = strongly agree)
   (a) Simon and I were on the same page.
   (b) Simon was team-oriented.
   (c) Our team worked fluently together.
   (d) Our team's fluency improved over time.
   (e) Simon was responsive to my actions.

(f) Simon listened to me.

(g) Simon talked over or interrupted me.

(h) I could tell whether or not Simon heard me.

(i) I could tell whether or not Simon understood me.

(j) I had to spend time waiting for Simon.

(k) Simon had to spend time waiting for me.

(l) There were awkward moments in the interaction.

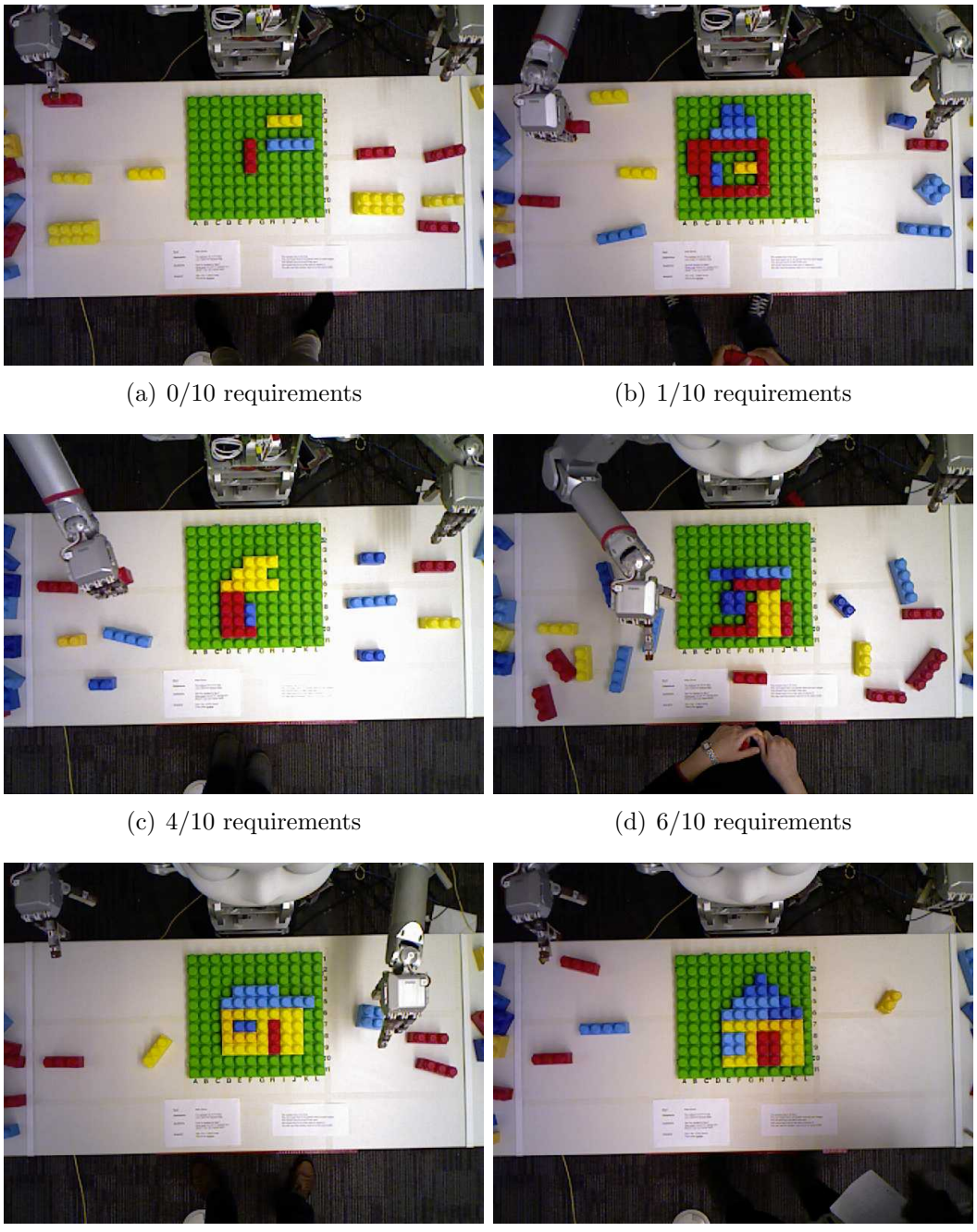
6. (Open-ended) Please provide a critical review of Simon as a team member.

9.4 Results

Overall, the results showed that in the experimental condition, the dyad was more successful at the task and interacted more fluently. We also characterize the amount of multimodal concurrency exhibited by the dyad in this particular task setting.

9.4.1 Task performance

For each participant, we examined the final frame recorded from the overhead sensor and noted the number of task requirements satisfied by the constructed model. In order to consider a requirement satisfied, all parts mentioned in the requirement had to be built; for example, in order to satisfy "the roof height has to be greater than the door height," both the roof and the door needed to be built. Several participants did not complete all parts of the house. If there was ambiguity about whether a part was completed, we referred to the participant's open-ended response to the question, "Did you complete the task successfully?" for more information about which parts the participant considered completed. Otherwise, ambiguous parts were considered not completed.



(a) 0/10 requirements

(b) 1/10 requirements

(c) 4/10 requirements

(d) 6/10 requirements

(e) 8/10 requirements

(f) 10/10 requirements

Figure 45: Examples of final states from different participants in the experiment, as viewed from the overhead sensor.

We found that the total number of requirements satisfied in the final built model (out of a maximum 10) was higher in the experimental condition ($M = 8.46$, $SD = 1.66$) compared to the baseline condition ($M = 6.31$, $SD = 2.95$), $t(12) = 2.82$, $p = .02$. Participants in the experimental condition also took less time to complete the task ($M = 12.3$ mins, $SD = 3.4$) compared to the baseline condition ($M = 14.7$ mins, $SD = 1.2$), $t(12) = 2.90$, $p = .01$. Most participants opted to use the entire 15 minutes allocated; 2 participants opted to stop sooner in the baseline condition, and 6 participants did in the experimental condition. To ensure that the earlier stopping times did not lead to worse task outcomes, we also checked the number of satisfied constraints divided by the session duration for each participant and verified this "constraint efficiency" was also significantly different. Indeed, the rate of constraint satisfaction in the experimental condition was $M = 0.77$/min, $SD = 0.35$ compared to the baseline of $M = 0.19$/min, $SD = 0.11$, $t(12) = 8.14$, $p < .001$.

Interestingly, we found that differences in number of requirements satisfied were due more to fewer of the *human's* constraints being satisfied compared to the robot's. In the experimental condition, participants satisfied $M = 4.23$, $SD = 1.23$ of the robot's requirements and $M = 4.23$, $SD = 0.83$ of their own requirements (out of a maximum of 5 each). In the baseline condition, participants satisfied $M = 3.54$, $SD = 1.56$ of the robot's requirements and $M = 2.77$, $SD = 1.69$ of their own. This was in spite of being able to refer to their own requirements on a sheet of paper in front of them. Thus, this imbalance might indicate a weaker mental model of task requirements due to increased cognitive load of interacting, or a feeling of lack of control over the task.

9.4.2 Fluency

We investigated the robot's response delays to human turns from the system logs. Robot responses were taken to be any collaboration act committed to interaction

Table 14: Responses to Likert-scale survey questions

Statement	Experimental	Baseline	p
(1=strongly disagree, 7=strongly agree)			
Simon and I were on the same page.	4.38 (1.39)	3.46 (1.51)	.06
Simon was team-oriented.	5.15 (1.52)	3.92 (1.50)	.03*
Our team worked fluently together.	4.08 (1.38)	2.69 (1.25)	.007*
Our team's fluency improved over time.	5.46 (1.05)	4.62 (1.50)	.05
Simon was responsive to my actions.	5.77 (1.30)	4.62 (1.61)	.03*
Simon listened to me.	5.85 (1.07)	5.38 (1.61)	.20
Simon talked over or interrupted me.	3.08 (1.93)	2.77 (1.83)	.34
I could tell whether or not Simon heard me.	5.23 (2.00)	4.54 (2.15)	.20
I could tell whether or not Simon understood me.	4.92 (1.93)	5.23 (1.64)	.33
I had to spend time waiting for Simon.	5.00 (1.15)	5.85 (0.90)	.02*
Simon had to spend time waiting for me.	4.92 (1.44)	4.15 (1.28)	.08
There were awkward moments in the interaction.	5.69 (1.18)	5.77 (0.93)	.43
(1 = very slow, 7 = very fast)			
How did you find the pacing of the interaction?	3.00 (0.82)	2.46 (0.88)	.06

history that was not the first act in its thread. We subtracted the timestamp of each of these acts from the previous (human's) act in the thread. The robot's response delays were significantly shorter in the experimental condition at $M = 1.2$ secs, $SD = 2.1$, compared to the baseline of 5.3 secs, $SD = 7.6$, $t(427) = 10.52$, $p < .001$.

In the responses to survey questions, there were significant differences for the statements: "Simon was team-oriented" "Our team worked fluently together," "I had to spend time waiting for Simon," and "Simon was responsive to my actions" (see full results in Table 14). Thus, the differences in fluency due to increased responsiveness of the robot appeared to be consciously noticeable. Overall, however, the robot's pacing was still quite slow. Participants in both conditions rated the pacing on the slow side and felt that they had to spend time waiting for Simon. This could be attributable to the robot's slow speed of manipulation.

9.4.3 Balance of control

To further understand the reasons for the task performance differences, we also wanted to examine how actions were distributed between the human and the robot. We distinguished between two categories of action, speaking and manipulating. The robot behavior was extracted from system logs, and human speaking and manipulation segments were annotated by two coders. One coder annotated the data for all participants; these annotations were used for the analysis in this section. The second coder annotated data for four participants (two per condition), only for the purposes of calculating intercoder agreement. Agreement was defined as the percentage of time that the annotation matched between the two coders. The agreement for speech segments was 98.2%, and the agreement for manipulation segments was 92.2%.

Gesturing at or touching objects was counted as manipulation. All phases of arm motions (reach, stroke, retract) were included in manipulation segments. Additionally, we annotated the dialogue acts for the human speaking turns.

First, we found that the robot completed more collaboration acts in the experimental condition than the baseline. Normalizing the number of acts by session duration, the robot committed $M = 2.53$, $SD = 0.80$ manipulation acts/min in the experimental condition compared to $M = 1.81$, $SD = 0.55$ acts/min in the baseline condition, $t(12) = 3.41$, $p = .005$. The robot also committed $M = 4.54$, $SD = 1.29$ speech acts/min in the experimental condition compared to $M = 2.82$, $SD = 1.00$ acts/min in the baseline, $t(12) = 4.40$, $p = .002$.

However, the robot also spent *less time* manipulating and had *higher idle time* in the experimental condition, in spite of having completed *more actions*. The robot's idle time normalized by session duration in the experimental condition was $M = 0.45$, $SD = 0.06$ compared to $M = 0.28$, $SD = 0.07$ in the baseline, $t(12) = 10.61$, $p < .001$. The robot's normalized manipulation time in the experimental condition

was $M = 0.36$, $SD = 0.07$ compared to $M = 0.49$, $SD = 0.16$ in the baseline condition, $t(12) = 3.43$, $p = .005$. These results are of course due to the robot's concurrent execution in the experimental condition; the robot could sometimes speak and manipulate simultaneously or perform multiple manipulation actions simultaneously. The higher idle time indicates that the robot ran out of actions to perform, which could happen when there were no reachable blocks in the workspace and the robot had nothing more to communicate.

There were no significant differences in human behavior between the conditions in terms of time spent manipulating or speaking, or the number of speech acts. (We did not annotate the number of human manipulation acts, since humans often manipulated groups of blocks at a time). Compared to the robot, the human spent less time speaking and roughly the same amount of time manipulating. Combining data from both conditions, the ratio of human speaking time to robot speaking time was $M = 0.36$, $SD = 0.15$, and the ratio of human manipulation time to robot manipulation time was $M = 1.02$, $SD = 0.62$.

The differences in numbers of dialogue acts were relatively minor. Humans asked more polar queries in the experimental condition, $M = 0.45$/min, $SD = 0.43$ compared to the baseline, $M = 0.16$/min, $SD = 0.25$, $t(12) = 2.37$, $p = .04$. Since polar queries are thread openers, this may be interpreted as increased initiative in the speech channel due to humans being more confident about when it was appropriate to speak. The robot also asked more clarification requests in the experimental condition, $M = 0.68$/min, $SD = 0.40$ compared to the baseline of $M = 1.43$/min, $SD = 0.90$, $t(12) = 3.17$, $p = .008$. This is explained simply by CRs being the robot's most common dialogue act due to speech recognition errors, and the robot having more opportunity to speak in the experimental condition due to the possibility of concurrent action.

The overall consistency of human behavior across both conditions would lead

us to conclude that task performance differences were mostly due to differences in robot behavior. However, we emphasize that humans completed the bulk of the manipulation for the task, since the humans were more dexterous, had access to more blocks, and also controlled the allocation of blocks in the robot's workspace. Thus, the robot most likely did not contribute in the experimental condition by taking over the task, since it had limited ability to construct the model, but by improving the accuracy of the human's mental model.

Overall, the human was idle more often than the robot. Combining both conditions, the human was $M = 1.76$, $SD = 0.64$ times more idle than the robot. Human idle time was a combination of several types of behavior. The human could be actively listening to the robot and not manipulating due to either wanting to wait for more information before acting or due to being at full attentional capacity. Participants also sometimes spent a large segment of time staring down at the list of requirements, presumably quietly thinking about the task. The interaction structure did not encourage "thinking out loud" because the robot responded to all off-grammar utterances with clarification requests (although many off-grammar remarks were still made). In contrast, robot idle time was almost never due to cognitive bottlenecks (except for the occasional 1-second solving time), but rather due to resource bottlenecks.

9.4.4 Multimodal concurrency

Following our analysis of time spent speaking and manipulating, we were also interested in characterizing concurrency of speech and arm actions for each agent and across agents.

Table 15 compares time spent on each modality combination across the conditions, normalized by session durations. Trivially, the time the robot spent concurrently speaking and manipulating was significantly higher in the experimental condition,

when multiple collaboration acts could be parallelized. There was still some concurrency in the baseline, due to certain collaboration acts using the arm to point (e.g. "Can you remove this?" and "Is this the door?"). Concurrent speaking and manipulating accounted for $M = 0.10$, $SD = 0.03$ of the time in the experimental condition compared to $M = 0.03$, $SD = 0.02$ in the baseline, $t(12) = 12.56$, $p < .001$.

We observe that the human's self-concurrency was not significantly different across conditions. Further, the amount of self-concurrency exhibited by the human in both conditions was actually more similar to the robot's self-concurrency in the baseline condition than in the experimental condition; compare the data for RS–RM and HS–HM in Table 15. Basically, the ability to multitask by manipulating multiple objects while talking about other topics turns out not to be particularly humanlike — at least not this early on in the interaction, before practice effects are witnessed. In addition, the grammatical constraints, task rules, and novelty of the experience may have been a cognitive burden to participants, preventing them from acting more concurrently. On this dimension, robots may differ fundamentally from humans; attentional bottlenecks can pose less of a limitation, but physical speed and dexterity are inferior, so increased self-concurrency may still be preferable behavior.

The conditions also differed on cross-agent concurrency. The experimental condition saw increased concurrency of the human speaking over the robot's manipulation and vice versa. Simultaneous speech was also higher in the experimental condition, but this was due in part to a higher amount of speech in general. For both agents, speech was very infrequent relative to the maximum interaction duration of 15 minutes, and manipulation was much more frequent, occurring roughly half the time. To control for these differences, we also examined the likelihood of concurrency compared to random chance. For a pair of modalities M_1 and M_2, this was defined to be:

$$p(M_1, M_2) = \frac{t_{M_1 \cap M_2}}{t_{M_1} t_{M_2}} \tag{8}$$

Table 15: Multimodal concurrency as a fraction of session duration. H = human, R = robot, S = speech, M = manipulation

Combination	Experimental	Baseline	p
RS–RM	0.097 (0.028)	0.025 (0.018)	<.001*
HS–HM	0.025 (0.026)	0.017 (0.017)	.17
HS–RS	0.004 (0.003)	0.001 (0.001)	.002*
HS–RM	0.045 (0.020)	0.040 (0.015)	.22
HM–RS	0.058 (0.040)	0.033 (0.032)	.04*
HM–RM	0.204 (0.082)	0.232 (0.117)	.24

That is, we divided the time that modalities M_1 and M_2 overlapped, $t_{M_1 \cap M_2}$, by the product of the times spent on each individual modality, t_{M_1} and t_{M_2}. A value close to 1 would indicate that these modalities naturally overlap in interaction (similarly to if the agents were not even interacting, but conducting independent activities). A lower value would indicate that the agents were actively avoiding overlap for that modality pair, or the presence of additional bottlenecks.

The resulting values from this calculation are shown in Table 16. $p(HM, RM)$ was very close to 1 for both agents manipulating, as well as the human speaking over the robot manipulating. However, it was lower at 0.7 for the human manipulating while the robot was speaking. From our qualitative observations during the experiment and annotation process, humans did sometimes interrupt or stop their manipulation actions when the robot spoke. This could be due to: 1) the manipulation action no longer being needed after a change in goal state implied by the new information, 2) a deliberate intention to wait while gathering information in anticipation of a possible state change, or 3) a relatively greater cognitive difficulty of processing new auditory input while executing a manipulation action, in comparison to processing visual input while executing a speech action. For both agents speaking, the factor was the lowest, indicating a deliberate intention to avoid simultaneous speech.

Table 16: Cross-agent multimodal concurrency compared to random chance, aggregated across both conditions

	Robot speaking (RS)	Robot manipulating (RM)
Human speaking (HS)	0.16	0.93
Human manipulating (HM)	0.70	1.02

9.5 Summary

In this chapter, we demonstrate the application of CADENCE to a domain of situated collaboration. The robot and the human must collaborate to design and build a block model of a house together using spoken dialogue and physical object manipulation. Each agent only knows 5 constraints for the final model, and together they must satisfy all 10. Because of partial information and noisy sensing, the robot must take initiative to maintain common ground through discourse strategies such as justification, propositional repairs, clarification requests, and grounding queries.

To evaluate the effectiveness of CADENCE for human-robot collaboration, we conduct a between-groups experiment with 26 participants. In the baseline condition, the robot executes a single atomic intention at a time, mimicking the behavior of simple state-based approaches. In the experimental condition, the robot uses CADENCE to perform concurrent execution and interruptions of multiple intentions based on resource availability for multiple resource types. We hypothesize that CADENCE's resource model should enable the robot to respond more quickly and naturally, resulting in more fluent turn-taking and therefore better task performance.

Our results from this experiment showed that when using the turn-taking model in CADENCE, the robot was indeed more responsive, exhibiting shorter response delays. The robot was subjectively perceived as a better teammate by participants. Using CADENCE, the dyad also performed the task more successfully, satisfying more constraints in less time.

CHAPTER X

CADENCE: DESIGN AND IMPLEMENTATION

The organization of this thesis has alternated new developments within CADENCE with the experimental validation of specific design decisions, mirroring the order in which the research was conducted. Due to the iterative nature of this approach, however, certain elements of various stages of implementation are irreconcilable while others are compatible. This chapter is dedicated to detailing the current state of CADENCE by condensing prior descriptions of its constituents in one place for increased clarity, as well as providing guidelines for practical application.

The first half of this chapter summarizes the abstract requirements of a control architecture for fluent social interaction. These requirements strongly motivate design decisions within CADENCE. We then provide implementation detail for components of CADENCE, including Petri net execution, TPN process creation, and domain definitions. We end with a summary of CADENCE's iterative development.

10.1 Requirements of an interaction architecture

In developing the BEAT architecture, Cassell et al. posited the following four requirements for controlling embodied conversational agents [23]:

- multimodal input and output,
- real-time,
- understanding and synthesis of propositional and interactional information,
- and a conversational function model.

Not surprisingly, all of these requirements for virtual agents are also necessary for the control of social *robotic* agents. Turn-taking skills for spoken interaction fall within the scope of the last requirement, the conversational function model, but are also highly dependent on the rest of the requirements to succeed.

CADENCE is informed by past work in virtual agents in addition to findings from human psychological studies, linguistic analyses, and newer developments in interactive systems. (The name is, of course, a tribute to BEAT.) We highlight three additional requirements for interaction architectures, which are fundamental to CADENCE's design:

- incremental processing,
- interacting concurrent subsystems,
- and separation of domain content from behavior.

Next we briefly elaborate on each of these requirements within the context of HRI.

10.1.1 Multimodal input and output

Social interaction is conducted through many modalities of communication, including speech, prosody, gaze, gesture, facial expressions, posture, proxemics, touch, and instrumental physical action. Humans integrate multiple modalities of information to achieve understanding. This phenomenon is well illustrated by the McGurk effect, in which phonemes are interpreted differently depending on whether visual input of the speaker is available [72]. In unimodal interactive systems, the absence of additional modalities restricts the communicative competence of the system, as in speech-only call center dialogues or typed text interactions that may not use gesture for referring.

The computation of modality inputs and outputs are distinct problems. A challenge for interaction is the asymmetry of competence in one versus the other. For

example, generating a robot's gaze behavior is simpler and more reliable than perception of human gaze directions. CADENCE handles the modalities of speech, gaze, gesture, and manipulation for controlling the robot and for perception of the human.

10.1.2 Real-time

The notion of a "real-time" system is often relative; after all, motor control systems must often operate at 1 kHz or more for correct performance. Here, we mean simply that the robot should exhibit timely responsiveness to the human's actions, adhering to similar processing delays that humans expect from other humans. Practically, this usually means decisions within tens or hundreds of milliseconds; Cassell et al. characterized their real-time requirement as "sub-second." CADENCE executes in a loop of approximately 30 Hz to achieve real-time interaction performance. The importance of real-time responses is also highlighted in Nooraei et al.'s recent development of the DiscoRT architecture (Disco Real-Time) [85], which extends a prior framework for collaborative discourse planning that did not account for timing.

10.1.3 Understanding and synthesis of propositional and interactional information

Propositional information refers to the content of an interaction, used for understanding and generating meaningful speech acts and actions that progress a task. Interactional information refers to behavior that has a regulatory role in an interaction, such as gaze cues, backchannels, or beat gestures. The open-ended object play experiment in Chapter 6 used a controller that relied on interactional information but not propositional information.

In the latest version of CADENCE, interactional information is represented in the turn-taking regulation parameters for specific resource types, as well as the social attention module that automatically produces gaze behavior based on propositional content and floor state. Propositional information is encoded in the semantic lexicon.

10.1.4 Conversational function model

The conversational function model describes all parts of the turn-taking process, which we have defined to be seizing the floor, holding the floor, yielding the floor, and auditing the interaction partner. The floor is referred to specifically in a conversational context, but any resources shared with a human may apply. Seizing of available resources is driven by goal-oriented action as well as a regulatory factor representing a sense of obligation to continue the interaction. Resources are held by actions operating on them, until intent is lost or they must be yielded to the human. When the robot does not have the floor, it continues generating behavior such as gaze and backchannels in order to communicate understanding and engagement.

Cassell et al. describe the conversational function model as also including the processes of initiating engagement and disengaging, which are currently outside the scope of CADENCE.

10.1.5 Incremental processing

Incremental processing describes the continuous nature of understanding and generation in interaction. In observations of human interaction, Clark and Krych described human turn-taking as being characterized by continuous, ongoing signaling in both directions, rather than a strict alternating structure [30]. Incremental processing stands in contrast to batch processing, such as waiting for the endpoint of a segment of speech before processing the entire segment all at once. The advantage of performing incremental processing is increased responsiveness and faster interaction pacing.

Also related is the notion of integrating continuous *bottom-up* and *top-down* processing. These terms are used loosely in cognitive psychology to denote processing of low-latency sensory signals and features (bottom-up) versus higher-latency semantic integration and interpretation from priors and context (top-down). Both should

adhere to the real-time requirement, but top-down updates may acceptably be less frequent.

There are several ways that continuous processing is important to CADENCE. First, turn-taking is driven predominantly by continuously monitoring low-level signals such as voice activity and human hand trajectories. Results from incremental speech recognition and natural language understanding are buffered until it is an appropriate time to take a turn. The construction grammar of CADENCE supports incremental language understanding by continuously processing semantic units and dialogue acts before the human releases the floor. These semantic units can be merged into records across multiple turns for faster repairs, or used to control gaze behavior for increased transparency.

Incremental processing is also essential to the concept of minimum necessary information. In the autonomous Simon Says game of Chapter 3, incremental gesture recognition of the human allowed the robot to start or interrupt its actions as soon as MNI was reached, leading to a faster-paced game.

Continuous processing is also relevant on the generation side. An example is dialogue acts with referring expressions, which can have multiple possible surface realizations using different resource sets. For example, one can use the floor to say "the red two by two block," or use the floor and an arm to say "this one" with a pointing gesture. Resources and action progress are monitored continuously throughout execution. If resource availability changes mid-utterance, CADENCE can switch to an alternative resource set or fully interrupt if none is available.

10.1.6 Interacting concurrent subsystems

Social interaction is also characterized by interacting concurrent subsystems. Multiple agents act concurrently when collaborating with each other. A single agent exhibits

concurrency when performing multiple intentions or using multiple modalities of action for a single intention. Not only that, but these subsystems have dependencies on each other. If a robot and a human are side by side but operate completely independently, this can hardly be called an interaction; when they act concurrently while sharing resources, they become interacting concurrent subsystems within a larger system of dyadic collaboration.

Such loosely coupled subsystems are the basis of using timed Petri nets to model behavior. CADENCE subsystems include interruptible action processes for each modality, resource controllers for each resource type, and user models for each resource type, all with clearly defined interactions with each other. This modular framework allows subsystems to be developed relatively independently and enables straightforward creation of behavior Petri nets for any domain by connecting the relevant subsystems.

10.1.7 Separation of domain content from behavior

Domain-specific content describes propositional information and task models, which can be sourced through several techniques such as authoring, learning, or crowdsourcing. Ideally, this knowledge should have a clear separation from general-purpose social skills that automatically produce behavior. The purpose is to increase the transferability of social skills between domains and the ease of developing new domains of interaction. Also, if social skills are coupled to domain content in the behavior, social intelligence is arguably not represented in the interaction architecture itself but in the process of designing that behavior.

The Towers of Hanoi interaction controller from Chapter 5 is an example of a design that did *not* adhere to this requirement. The Petri net in Figure 22 on page 66 is tied directly to the interaction for this specific task and is not easily reusable for any other task.

The next version of CADENCE in Chapter 6 achieves a better separation, similarly to BEAT and approaches descended from it. Turn-taking has a separate generalizable model, but each turn is still based on a specific ad hoc specification of aligned speech, gaze, and gesture.

In the latest version of CADENCE, described in Chapters 7–9, modality behavior for speech, gaze, and motions are generated based on an underlying semantic form, and any behavioral temporal alignments are based on constraints on the underlying semantic units. The authoring of semantic and task information within the semantic lexicon is clearly separated from turn-taking and the production of behavior.

10.2 Implementation guidelines

This section provides more detail about how to implement a behavior controller using CADENCE. We describe how TPNs are executed, constructed, and connected within the framework. We then describe the processes for developing a new domain using the framework, and for developing new behavioral skills to augment the framework.

10.2.1 Petri net execution

Within CADENCE, timed Petri nets are used for generating behavior for turn-taking and action execution. Several libraries are available for Petri nets, but they are also straightforward to implement, which we have done to better integrate with our existing middleware.

Algorithm 2 provides pseudocode for the main behavioral loop, running at approximately 30 Hz. At each cycle, the *dispatcher* is updated based on the output of the dialogue manager. As a reminder, the dispatcher update is responsible for spawning new action tokens at entry points to action processes in the TPN, following temporal constraints within intention hierarchies. Then each transition t is visited and its guard function $\mathcal{G}(I)$ is run on t's input place set I. A transition is *enabled* if the guard was inactive the previous cycle but is active the current cycle. An activated

Algorithm 2 TPN-based behavior execution

```
update system clock C(i, τ) → τ'
update dispatcher
for all t_j in transition set T do
    if guard G_j(I)_{i+1} = 1 then
        if G_j(I)_i = 0 then
            enable t
        end if
        run firing function F_j(M, I, O, τ') once
    else
        if G_j(I)_i = 1 then
            disable t
        end if
    end if
end for
```

guard allows the firing function to fire once. This may not necessarily change the graph marking immediately, but through subsequent cycles, will allow the function to continue running until the marking does change. If the guard function is not active but was active the previous cycle, the transition is disabled.

10.2.2 Instantiating TPN processes

A core principle of the behavior model in CADENCE is interacting concurrent subsystems that are also TPNs, which are easily extensible or reusable between controllers.

10.2.2.1 Action processes

CADENCE uses the same basic Petri net process template for different modality actions, such as speech, motion generation, or grasping. This is shown in Figure 34 on page 121. Modality processes differ in how they define how an action should *start*, *pause*, *resume*, *hold*, *stop*, or when it is considered *finished*. These commands are used inside of the transition firing functions for that action process.

- *start(action)* – runs once to setup and start the action. This is only run after action preconditions are satisfied, including the action's required resources being owned by the action process's resource controllers.

- *pause(action)* – temporarily suspends the action execution. This is used for hesitating at the onset of a resource conflict to communicate awareness of the conflict.

- *resume(action)* – resumes a suspended action; may be the same behavior as starting the action over again. This is used to continue the action if required resources are no longer in conflict because the human backed off within a time window.

- *hold(action)* – runs continuously to execute the action. Also continuously monitors the action's progress (for minimum necessary information) and state of completion.

- *stop(action)* – aborts the action fully; may be the same behavior as pausing the action. This is called when resources have become unavailable or contested for too long, exceeding the robot's conflict tolerance, or alternatively when the action is no longer intended for the task.

- *finished(action)* – returns the finishing condition for the action.

Algorithm 3 provides pseudocode for implementing this specification for one modality, speech.

The parameters that control action processes are self-interruptibility, which describes whether or not that modality can interrupt (including both pausing and stopping), and the timing windows for pausing and resuming.

10.2.2.2 Turn-taking models

A turn-taking model exists for each *resource type*. Examples of *resource types* are the speaking floor, spatial regions, or objects. The number of resources may or may not be bounded. In our dyadic experiments, there was always a single speaking floor, but other social settings may require more, such as when a four-participant interaction

Algorithm 3 Example of implementing the action process specification for the modality of speech.

```
function START(SpeechAct)
    setup N phonemes
    index_phoneme ← 0
    start speech synthesis
end function

function PAUSE(SpeechAct)
    stop speech synthesis at word boundary
end function

function RESUME(SpeechAct)
    if time_paused > threshold then
        index_phoneme ← 0                    ▷ start again from the beginning
    end if
    start speech synthesis
end function

function HOLD(SpeechAct)
    if last phoneme finished then
        index_phoneme ← index_phoneme + 1
    end if
end function

function STOP(SpeechAct)
    stop speech synthesis
end function

function FINISHED(SpeechAct)
    return index_phoneme = N
end function
```

breaks up into two dyadic interactions [69]. We also fixed the number of spatial regions by discretizing the robot's workspace. Our object model was unbounded because of the nature of the perception used, in which objects could enter and leave the scene and could not be uniquely identified; however, one can also conceive of a task in which all of the objects are known in advance, resulting in a fixed number of object tokens in the TPN.

The turn-taking model for a given resource type requires the definition of three components: a resource monitor, a user process, and a robot resource controller. These are defined and discussed in depth in Chapter 7. To summarize:

- **Resource monitor**. The resource monitor is the generalized version of the parameterized floor regulator from Chapter 6. The responsibility of the resource monitor is to decide how to assign resources between the robot and user. The relevant parameters for the resource monitor are conflict tolerance, describing how long to tolerate a resource being in conflict before the robot fully backs off; user-interruptibility, describing whether it is permissible to barge in to seize resources from the user; and deep-interrupt time, the amount of time wait for a resource before such a barge-in can trigger. (In the final version of CADENCE, lapse avoidance is performed in the dialogue manager instead.)

- **User process**. The user process for a resource type essentially functions as the counterpart for the robot's resource controller, but is based on perception rather than controllable state. The user process must define perception (or simulation) for *holding(resource)*, *signaling(resource)*, and *yielding(resource)*. Two examples are provided in Table 17. Holding indicates current resource ownership, whereas signaling is used to request resources. These may sometimes be difficult to distinguish; for example, in our floor model in Chapter 6, these were both based on voice activity. However, one can imagine a more sophisticated

Speaking floor	**Holding**: incremental ASR buffer contains partial results **Signaling**: pitch exceeds a threshold for a few hundred ms **Yielding**: ASR reached a grammatical endpoint, no voice activity
Spatial regions	**Holding**: an arm is over the region **Signaling**: a hand is moving towards the region **Yielding**: arms are not over the region, or are moving away

Table 17: Examples of implementing the user process for two resource types, the speaking floor and spatial regions, which were used for the block collaboration experiment in Chapter 9.

floor signaling model that used more subtle cues such as mouth openings, breath intake, or hand gestures as described in the literature [35, 86]. For spatial regions, these are more distinctly different: occupying a spatial region is holding it, but reaching towards a spatial region is signaling for it.

- **Resource controller**. The robot resource controller for a type is the most straightforward, since it behaves the same way for all resource types. The resource controller seizes and releases resources based on the status of the action. It simply must be created and then connected to all relevant action processes that require the resource type to execute.

Table 18 provides an example of defining each of these components for a complete interaction that includes speech and manipulation.

10.2.3 Domain specification

When a new domain of interaction is being developed, the following are components that must be defined:

1. **Items in the semantic lexicon.** These are the objects, features, actions, tasks, and other semantic primitives that the robot must be able to reason and talk about. When implementing a collaborative domain from scratch, it may be helpful to have two humans perform the desired task in order to develop a

Resource type	speaking floor (1, bounded)
Robot controller	TTS action process
User controller	ASR/VAD user process
Monitor	interrupt self = true interrupt user = false conflict tolerance = 300 ms

Resource type	spatial regions (5, bounded)
Robot controller	motion action process
User controller	skeleton tracking user process
Monitor	interrupt self = true interrupt user = false conflict tolerance = 500 ms

Resource type	table objects (unbounded)
Robot controller	grasping action process
User controller	block tracking user process
Monitor	interrupt self = false interrupt user = false conflict tolerance = 0 ms

Resource type	robot DOFs (37, bounded)
Robot controllers	motion action process grasping action process gaze action process
User controller	(none)
Monitor	Robot's internal DOF arbitration mechanism. – Gaze receives the lowest priority. – Other processes are first-come, first-served. – Gaze interrupts to yield resources to other processes.

Table 18: Example details for implementing the resource model specification of an interaction, organized by resource type. Each resource type has one or more robot controllers. If the resource is used in turn-taking, the resource type also has a user controller.

minimal set of semantic primitives and dialogue acts. An example of a complete semantic specification for a task is provided in Section 9.1.1 on page 154.

2. **Collaboration act processors.** Processors operate in parallel to interpret and generate collaboration acts. For each act, decide if the robot must interpret it from the human, generate the act, or do both. Define the semantic signatures for these acts. Most collaboration act processors should be general-purpose and not dependent on any particular task. It may be additionally necessary to include a domain-dependent processor dedicated to continually generating goal-oriented actions to progress the task. Given the set of processors with a generation component, define their *regulators*, which determine valid times for them to be run. Regulators were originally defined in Section 8.8.1 on page 148. For example, second pair parts should occur soon after the first pair parts they are responding to, but acts that start new threads should wait longer. A list of all of the processors and regulators used for the domain of collaboratively building a block house is available on page 164.

3. **Arbitration for interpretations and options**. Collaboration act processors act in parallel and may produce competing interpretations and options. In the block collaboration experiment in Chapter 9, interpretation arbitration was based on maximizing coverage of words in the input speech buffer, and option arbitration prioritized second pair parts before first pair parts. These are relatively general-purpose, but depending on the task, domain-dependent arbitration mechanisms may also be necessary.

4. **Parameters appropriate to the interaction context.** Relevant factors for deciding parameter values may include task roles, familiarity or relationship, relative status, and cultural norms. For example, the robot may want to be more dominant if it is teaching than if it is learning. Specific values may be set

Parameter	Value	Potential scenarios
Interrupt self	true	service role, where status is lower
		providing information when goal is unknown
		learning, where actions have high rate of failure
	false	maintaining control as a teacher or tour guide
Interrupt user	true	stopping a costly mistake
	false	acting mostly as a tool
Conflict/overlap tolerance	high	familiar relationship or culture
	low	formal setting or culture
Regulatory delays	high	novice human
(response, initiate, lapse)	low	expert human
		task where efficiency is important

Table 19: Possible considerations when choosing turn-taking parameters

based on literature in psychology or linguistics, annotating public corpora or one's own data in the domain, or piloting the interaction with a small number of participants. Section 6.3.1 on 100 describes parameters for an active versus a passive robot. We also highlight some hypothetical examples in Table 10.2.3.

10.2.4 Iterative approach

In this research, we employed an iterative approach to development of the robot's social skills.

1. **Identification of a phenomenon**. Hypotheses about interaction phenomena can be based on shortcomings of a previous iteration of the system, psychological studies about cognitive or social phenomena, linguistic theories and corpora analyses, or even pedagogy of performance art and character animation.

2. **Modeling of the phenomenon**. In this thesis, the relevant phenomena pertained to temporally extended behavior, so computational modeling was conducted primarily through the vehicle of timed Petri nets. TPNs were used for modeling, controlling, and simulating interacting concurrent subsystems. Designs of new Petri net processes can be developed through the composition of

established workflow patterns [117]. For example:

- Action processes follow the patterns of *sequence* and *interruption handling*.
- Multiple modalities of action processes follow *parallel* execution.
- Resources shared between the robot and the user form a *mutual exclusion*.
- The dispatcher enables *synchronization* across modalities.

3. **Evaluation of the model**. Evaluation is best performed through objective and subjective metrics in user studies. Simulation can augment such evaluations by characterizing system dynamics in broader situations. In conducting user studies, shortcomings of the model become apparent for the next iteration of the system.

The motivation for this iterative approach is the development of general-purpose social behavior skills that are compatible and persistent across multiple domains. It is equally possible to investigate individual phenomena in an isolated fashion, but the goal is to increase the holistic social intelligence of the robot, which requires their combination and integration.

To illustrate this approach, Chapter 5 described the first iteration of CADENCE. We identified the phenomenon of action interruptions and implemented it within a domain-specific timed Petri net. We evaluated the model in a domain of collaborative Towers of Hanoi, finding that the robot was sometimes too passive when interrupting itself all the time.

In the next iteration in Chapter 6, we sought to address that phenomenon with a better balance of seizing and yielding. We developed a floor regulation model based on parameters from linguistics and theater about dominance and social competence. Our evaluation in an open-ended object play domain validated the model, but also exposed a shortcoming in how cross-modality interruptions were handled.

In the iteration of CADENCE in Chapter 7, that phenomenon was addressed by generalizing the turn-taking model so that it could be used and repeated for multiple resource types. We evaluated that version in a domain of collaborating to design and build a block house. The result was a substantially more general formulation of turn-taking that subsumed the capabilities of previous iterations.

CHAPTER XI

CONCLUSION

We have described a framework for embodied, multimodal turn-taking that is characterized by seamless exchange of shared resources. The framework features additional timing parameters for further control over the interaction dynamics. In developing this framework, our methodology has been an iterative one of building autonomous robot behavior controllers and studying their resulting dynamics with human users, the results of which inform the next round of system implementation.

The result of our efforts has been CADENCE, the Control Architecture for the Dynamics of Embodied Natural Coordination and Engagement. CADENCE integrates turn-taking, action execution, and human modeling using a timed Petri net representation. To support reaching common ground with a human through situated multimodal interactions, CADENCE also includes a semantically rich general-purpose dialogue system. Through the development of CADENCE, we have enabled a social humanoid robot to interact fluently with humans using speech, gesture, gaze, and manipulation. We have evaluated the robot's turn-taking behavior in a variety of interaction scenarios encompassing dialogue and physical collaboration.

In Sections 11.1–11.4, we revisit the contributions of the thesis. This is followed in Section 11.5 by a summary of future work enabled by these contributions.

11.1 Factors important to multimodal turn-taking

Turn-taking has most commonly been studied in speech systems in which the primary concern is the speaking floor. In this thesis, we have additionally identified and modeled essential phenomena that stem from turn-taking in a situated, multimodal setting.

The first is the principle of minimum necessary information (MNI). Our initial studies in the domain of "Simon says" have underscored that turn-taking is hardly a simple matter of cue recognition, but driven primarily by semantic understanding. This was further reinforced in our analyses of open-ended object play, in which the robot's modality states did not predict when the user would seize the floor. Motivated by the importance of information flow in interaction, our dialogue system is structured around clusters of adjacency pairs called threads and supports faster turn-taking pacing through repairs of individual semantic units.

Another factor that we have formalized is the conception of turn-taking as the exchange of bottlenecking resources. That is, resources are consumed by modality actions to achieve intentions, and the surface alignment of modalities results in part from resource availability. When multiple modalities are needed to achieve a single intention, they are synchronized or jointly interrupted. When multiple modalities are executed concurrently to achieve separate intentions, they are independently interruptible.

Finally, we have modeled key timing parameters in this model that are critical to the dynamics of the interaction. By controlling the timing by which a robot seizes and yields resources, the robot can exhibit turn-taking behavior that suits a range of different social roles.

11.2 An integrated interaction architecture

To validate our turn-taking model, we have implemented it within the context of an integrated architecture for social human-robot interaction. CADENCE integrates the social modalities of speech, gesture, and gaze with manipulation on a physical robot. The architecture design follows the lessons imparted by previous advances in embodied conversational agents, spoken dialogue systems, and human-robot interaction. Gaze at the human or objects is controlled based on turn-taking floor state.

Gaze, speech, and gesture are appropriately synchronized according to underlying semantics. Voice activity is used to monitor overlapping speech. Head gestures are automatically generated based on dialogue act function.

CADENCE also presents novel conceptualizations of core social cognition modules. Atypically, turn-taking is positioned as a central, explicit process in the architecture, contrasting with approaches where turn-taking is dismissed as emergent behavior. Like a periodic pulse, this process drives the cyclical patterns in the dyad's behavior by bridging the robot's action execution with the perceptual model of the human user. CADENCE also models a dynamic awareness of resources shared between the human and the robot, generating real-time action interruptions in response to resource availability. In addition, CADENCE continues to generate behavior when the robot does not hold resources, auditing the interaction partner in order to maintain engagement and transparency.

11.3 New domains of human-robot interaction

In this work, we have also demonstrated four novel domains of autonomous social human-robot interaction, each with its unique interaction dynamics.

Our "Simon says" domain captured two phases of interaction. The faster-paced game phase used speech, gesture, and gaze, and aimed to minimize response delays. The speech-only negotiation phase was used to switch roles. The "Simon says" game was later used by other HRI researchers to investigate social skills in children with autism [36].

We also demonstrated two domains involving physical collaboration between a human and a robot. In the Towers of Hanoi collaboration, both the human and the robot could move the puzzle pieces to achieve the goal, and the robot could request the human for actions. In the block house task, the human and the robot had to reach a common mental design and physically construct it after each starting with

partial information. The former domain showcased mostly physical joint action, and the latter expanded this scenario to include situated dialogue focused on information transfer and repairs.

In our open-ended object play domain, we demonstrated an interaction with speech, gesture, gaze, and manipulation that featured a dynamic that was more similar to human-human conversation. Because situated natural language understanding is still an open research problem, utterances in human-robot dialogue tend to be short. The technique of generated an artificial language can be used to elicit human responses with natural turn-taking timing without the limitations of speech technologies or the influence of domain semantics.

11.4 Experimental results about turn-taking

At each iteration of CADENCE, we have stopped to perform evaluations of the robot's latest turn-taking behavior. These results have offered insights into the benefits of improving turn-taking as well as the nature of how humans interact.

In both of our physical collaboration scenarios, our results in user studies showed that using our turn-taking model resulted in shorter task completion times. In the more difficult task of designing a block house with 10 constraints, dyads using our turn-taking model also satisfied more of the constraints. On further analysis, we conclude from our results that appropriate turn-taking leads to more efficient resource usage and maintains consistency between the human's and the robot's mental models.

Results also showed that participants subjectively rated our turn-taking model as producing more fluent interactions. In our final experiment, participants using our turn-taking model viewed the robot as less awkward and more team-oriented, and felt they had to wait less on the robot. These subjective perceptions were supported by quantitative phenomena in the data such as shorter response delays from the robot and increased cross-modality concurrency between the human and the robot.

In our investigations, we have also found that that people will increase their initiative in turn-taking when the robot is more passive. In the Towers of Hanoi collaboration, people performed more manipulation actions when the robot interrupted its own actions. In the open-ended object play domain, people also spoke more to fill silences when the robot had a passive parameter setting. The adaptive, regulatory nature of human turn-taking appears to generalize across modalities and resources.

11.5 Future work

The groundwork laid by CADENCE lends itself to several other interesting lines of inquiry.

11.5.1 Behavioral extensions

The clearest and simplest additions to CADENCE would be to support new modalities and resource types. For one, CADENCE does not encapsulate any notion of mobility, since it was demonstrated only on a stationary upper-torso humanoid robot. With the increased physical competencies of a mobile manipulator, turn-taking dynamics could change substantially. How should proxemics and social navigation be integrated into the robot's reasoning and control of spatial resources?

In addition, CADENCE could leverage more information from the human's modalities to improve turn-taking via the user process. For example, one addition would be to monitor the human's visual attention, which should be a resource required for gesturing. Another would be to leverage prosody in the human speech signal, which can be a strong indicator in disambiguating intent to seize or yield the floor [73].

11.5.2 Discourse strategies

Our work in dialogue has only scratched the surface of what is possible to make our robots more transparent and intelligent machines. We should strive to implement new categories of discourse strategies that can be automatically generated. This requires

additional natural language functions that must be supported by complementary general-purpose reasoning systems.

For example, a natural multimodal behavior exhibited by collaborating humans is to narrate actions as they are performed, which strengthens common ground state. With reasoning systems and natural language that support temporal logic, action, and consequence or causality, the robot could also talk about its actions in such a manner and understand the human doing so. This would increase transparency about intentions in both directions.

Multi-act turns also warrant more investigation. We used one multi-act turn in our block collaboration experiment, in which the robot could generate a rejection followed by a justification (comprising multiple inform acts). Multi-act turns were less prevalent in our domains of study but pose significant research challenges for turn-taking. When should a robot self-interrupt or barge in if either agent is able to take very long turns? Some potentially relevant domains could be teaching or learning interactions, or giving a live demonstration to sell a product. Also, it is easy to position robots as passive helpers, but are there situations where it is appropriate for a robot to barge in, take initiative, or have a forceful personality?

11.5.3 Larger-scale turn-taking factors

This thesis has focused primarily on micro-scale factors that sum together to produce natural turn-taking. In addition, turn-taking timing is influenced by larger-scale factors, such as adaptive pacing. Within a single interaction, human dyads can exhibit vastly different pacing, with certain segments having shorter and shorter delays and others having sparser turns. Rather than having static parameters, it may be more natural to allow contextually dependent parameter settings.

Related to dynamic pacing is the notion of integrating a mechanism for estimating the cognitive load of the human. In our experiments, we observed that humans do

not always spontaneously parallelize actions across modalities, which could be due to attentional bottlenecks. The robot could control its own concurrency to better match the human's cognitive load. This could be especially relevant in a difficult task, but one in which the dyad improves with practice. Appropriate turn-taking may mean something different when the task is novel for the human than when the human is a well-practiced expert.

Another parameter that could benefit from a more sophisticated model is the interruptibility of oneself or the user. Currently, interruptibility is binary. An uninterruptible robot can test human patience with long turns, but a robot that interrupts itself at the slightest human action ends up being overly passive. The same extremes exist for barge-ins. A robot that barges in every turn is highly annoying, but if the robot never barges in when the human acts for extended amounts of time, the robot contributes nothing. It would be interesting to have, for instance, a model that gradually increases the chance of barge-ins the longer that the user monopolizes resources. This could represent the "frustration level" of the robot.

Finally, it is important to understand turn-taking within a larger context of lifelong social interaction. Humans engage in turn-taking styles differently with different people depending on their relative statuses and their long-term relationships. An overly familiar turn-taking style could offend strangers, and a distant one could alienate friends. A social robot that finds itself in a diversity of roles and relationships would benefit from computational support for these differentiations.

11.6 Final remarks

I hope that you, the reader, have come away with newfound respect for multimodal turn-taking, a process you may have taken for granted since you were a normally developed two-year-old child. In fact, the whole of turn-taking has much more depth than any individual behavioral cue, any nod or glance [23] or thoughtful pause. It

describes the meeting of cognition and behavior within a mind, and the meeting of minds within an interaction.

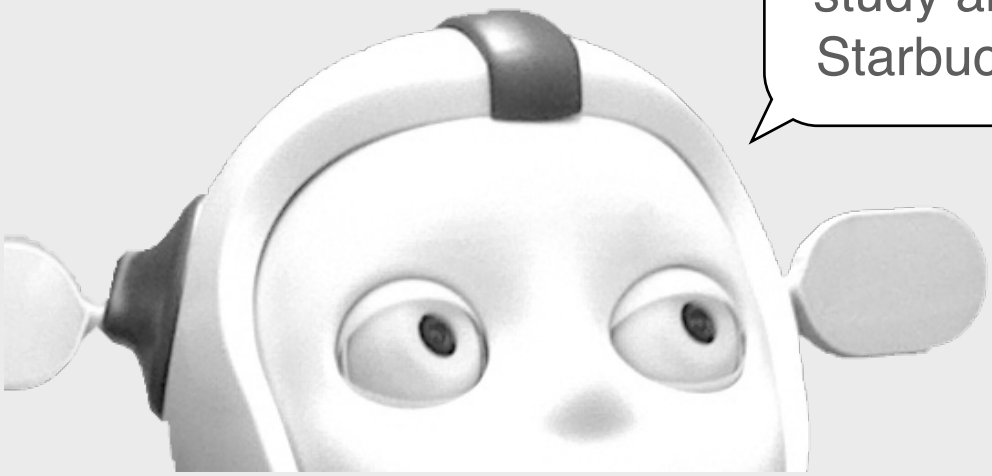
To any future generations of social roboticists who may be reading, I also request: make robots *do* things, make them *understand* things. It can be a great temptation as we conduct science to whittle systems down to their smallest constituents in the name of experimental control. Robots play back a single scripted behavior supposedly so that it can be unpolluted by other factors. Or, they become data collection devices for statistically beautiful results that never seem to make their way back to reality. Instead, we should build greater and greater systems. Only in so doing will we one day have robots who are worth taking turns with.

APPENDIX A

MATERIALS FOR SIMON SAYS STUDY

- Participant recruitment flyer
- Experimenter protocol
- Post-study questionnaire
- IRB consent form

Human-Robot Interaction Study



Interact with Simon the Robot and contribute to robotics research at Georgia Tech! Visit the website below to sign up.

simontherobot.com/study

Place: CCB 2nd floor, RIM Center Robotics Lab
Dates: Jan. 17–21 and Jan. 24–28, 2011

Socially Intelligent Machines Lab at Georgia Tech
simon@mail.gatech.edu

"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study
"Simon says" study simontherobot.com/study

Simon Says checklist

Meka server

1. Go to the RTPC and open up a terminal, and run: m3rt_server_run
2. In a another terminal window, cd ~/mekabot/m3sim/trunk/python/scripts and run ./m3sim_demo_c6_animation.py

External modules

- FaceAPI (# On Windows Machine)
- Open up two terminals, and go to {Program Folder Name/Release}: cd C:\Program Files\SeeingMachines\FaceTrackingAPI 3.1\Samples\Console\Release
- RunICE_FaceCam External_FaceCam 127.0.0.1 10110 // This runs External camera
- RunICE_FaceCam Robot_FaceCam 127.0.0.1 10120 // This runs Eye camera
- The images from the external camera and the eye camera (and two text files containing corresponding head poses) will be stored in the folder C:\Program Files\SeeingMachines\FaceTrackingAPI 3.1\Samples\Console\Release\data\"name specified"

- Kinect (# On New Linux Machine)
- Log on as siml (Passwd: LabPassword)
- Open up a terminal
- Go to /home/siml/projects/ExternalSensorLog/Vision/bin
- Run ./recordICE Kinect 127.0.0.1 10010

Synchronize time

1. On Unix/Mac, sudo ntpdate time.nist.gov
2. On Windows:
    1. Go to the Control Panel > Date and Time > Internet Time.
    2. "Click the Change Settings…" button.
    3. Select time.nist.gov from the dropdown.
    4. Click the "Update now" button.

Internal

- Speakers
    1. Connect the speakers to the new Mac Pro.
    2. Turn the speakers on by rotating the silver knob until the light is visible.
    3. On the Mac Pro, go to System Preferences > Sound > Output. Set the output device to "Headphones."
    4. Set the volume to about 90% of the bar.
- Microphone

1. Connect the mic base to the new Mac Pro.
2. Turn the mic base on by hitting the button.
3. On the Mac Pro, go to System Preferences > Sound > Input. Set the input device (varies depending on the mic).

C6 Controller

1. From the new Mac Pro, launch momotaz.datacollection.DataCollectController
2. In “Main Motors Enabler” window:
    1. Click the “Send data” checkbox.
    2. Click the “all” presets button.
    3. Click the “Update” button.
3. In the “Logging” window:
    1. Type the subject id and hit enter.
    2. Click the “init external” button.
    3. Verify “Finished initializing connection” output in the terminal window.
    4. Make sure the “Keyboard Events” window is visible.

Experiment

1. Have subject sign consent form.
2. Walk through protocol with subject.
3. Give mic to subject. Make sure it’s turned on.
4. Lift the e-stop.
5. In the C6 controller “Logging” window, click the “start log” button.
6. Double check that iMovie has actually started importing.
7. Tele-operate through the “Keyboard Events” window, starting with “g”.
8. When finished, click the “stop log” button.
9. Double check that iMovie has actually stopped importing.
10. Have subject do survey on iMac.
11. Give subject reward.
12. Have subject sign reward receipt form.

Subject instructions

You’re going to play repeated games of “Simon says” with the robot. The game has a leader and follower. We call the leader “Simon.”

As a refresher: in the game, the leader (a.k.a. Simon) can say “Simon says, do this.” After this the follower should do the same thing. But if he says “Do this” without saying “Simon says,” then the follower should do nothing; if he does something he loses. For example, if the robot is playing the leader and says, “Simon says, wave your arm,” then the follower should wave his arm. If the robot says, “Wave your arm,” the follower should not wave his arm, or else he loses the game.

There are 5 things you can do: flap your arms like a bird, bow, play air guitar, shrug your

shoulders, or wave your arm.

At any time, you can negotiate with the robot who gets to be the leader next. You can ask, “Can I play Simon now?” or “I want to play Simon now” or the robot may say that. The response from either of you can be yes or no. You can do this at the end of a game or in the middle of a game.

Note that the follower doesn’t lose the game if he can’t do the action, only if he does the action when he’s not supposed to. The robot can actually only do 5 things for this: bow, wave, shrug, flap his arms like a bird, and play air guitar. You can let him start first if you want to get used to it.

When the robot loses, say “You lose” or “I win!” If you notice yourself lose, you can say “I lose” or “You win!” Or the robot will notice it and say something.

We’ll have you do this interaction continually and stop you after (2?) minutes.

1. Simon says

*** 1. Enter your info:**

Name

Major

Gender

*** 2. Do you have prior experience with robotics?**

Yes

No

Describe (optional):

*** 3. Do you have prior experience with the game "Simon says"?**

Yes

No

*** 4. Do we have permission to use images of you in academic publications or conference presentations?**

Yes

No

*** 5. On a scale from 0 to 10, assuming a human gets a score of 10, how would you rate...**

... Simon's understanding of your spoken language?

... Simon's use of spoken language?

... the smoothness of the interaction with Simon?

*** 6. How helpful to the interaction did you find...**

	1 (Counter-productive)	2 (Unhelpful)	3 (Neither helpful nor unhelpful)	4 (Somewhat helpful)	5 (Very helpful)
... the content of Simon's speech?					
... Simon's head motions?					
... Simon's gaze?					
... Simon's arm and hand gestures?					

*** 7. Rate your agreement or disagreement with each statement.**

	1 (Strongly disagree)			4 (Neutral)			7 (Strongly agree)
Simon spoke at appropriate times.							
Simon moved at appropriate times.							
The interaction with Simon was interesting.							
I was fully engaged in the interaction with Simon.							

8. Any other comments, or any suggestions on how to make the interaction more natural? (optional)

Done

CONSENT DOCUMENT FOR ENROLLING PARTICIPANTS IN A RESEARCH STUDY

Georgia Institute of Technology
Project Title: Human-Robot Contingency and Turn-Taking
Protocol and Consent Title: 08/27/09v1
Investigators: Dr. Andrea Thomaz, Crystal Chao, Jeff Kiser, and Jinhan Lee

You are being asked to be a volunteer in a research study.

Purpose:
The purpose of this study is to learn more about robot technology designed to communicate with people using socially interaction.

Procedures:
If you decide to participate in this study, your part will involve:

- The study will take between 30 minutes and one hour.
- You will first be given an introduction by the experimenter. During this introduction the experimenter will explain the robot technology that will be used in the study.
- You will then be asked to socially interact with the robot technology. For example, our robot may try to get your attention or talk to you.
- We will collect audio and video data from your session. These tapes will remain confidential, and will not be distributed in any way or used for any purpose other than our analysis. The tapes will be erased after the study is finished.
- Finally, after the experiment is over, you will have the opportunity to ask questions and learn more about the goals of this research if you want to.

Risks or Discomforts:
The risks involved are no greater than those involved in daily activities such as holding conversations with people.

Benefits:
We hope that someday robots will be able to initiate and hold conversations with people. In the meantime, you may benefit from being in this study if you enjoy interacting with new technology and learning about the state of the art robotics research going on at Georgia Tech.

Compensation to You:
There is no compensation for participation.

Confidentiality:
The data collected about you will be kept private to the extent allowed by law. To protect your privacy, your records will be kept under a code number rather than by name. Your records will be kept in locked files and only study staff will be allowed to look at them. Your name and any other fact that might point to you will not appear when results of this study are presented or published.

The video and audio tapes collected as part of the evaluation will be stored in a locked cabinet in Dr. Thomaz's office. Dr. Thomaz and the students involved with the project being evaluated will have access to these tapes for the purpose of analyzing the human-robot interaction. Once the research project is completed, the tapes will be erased.

To make sure that this research is being carried out in the proper way, the Georgia Institute of Technology IRB may review study records. The Office of Human Research Protections may also look over study records during required reviews.

Costs to You:
There are no costs to you, other than your time, for being in this study.

In Case of Injury/Harm:
If you are injured as a result of being in this study, please contact Principal Investigator, Dr. Andrea Thomaz, at telephone
(404) 385-3365. Neither the Principal Investigator nor Georgia Institute of Technology has made provision for payment of costs associated with any injury resulting from participation in this study.

Participant Rights:
- Your participation in this study is voluntary. You do not have to be in this study if you don't want to be.
- You have the right to change your mind and leave the study at any time without giving any reason and without penalty.
- Any new information that may make you change your mind about being in this study will be given to you.
- You will be given a copy of this consent form to keep.
- You do not waive any of your legal rights by signing this consent form.

Questions about the Study:
- If you have any questions about the study, you may contact Dr. Andrea Thomaz at telephone (404) 385-3365.
- If you have any questions about your rights as a research subject, you may contact:

Ms. Melanie Clark, Georgia Institute of Technology
Office of Research Compliance, at (404) 894-6942.

Your Consent:
If you sign below, it means that you have read (or have had read to you) the information given in this consent form, and you would like to be a volunteer in this study.

_____
Participant Name (printed)

_____ _____

Participant Signature Date

_____ _____

Signature of Person Obtaining Consent Date

APPENDIX B

MATERIALS FOR TOWERS OF HANOI COLLABORATION STUDY

- Experimenter protocol
- Post-study questionnaire
- IRB consent form

HRI 2012 turn-taking study protocol

Meka server

1. Go to the RTPC and open up a terminal, and run: m3rt_server_run
2. In a another terminal window, cd ~/mekabot/m3sim/trunk/python/scripts and run ./m3sim_demo_c6_animation_new.py

External modules

- Vision (On old Mac Pro)
- In Maya's account (lol), cd to ~/projects/c6m/maya/vision.
- Run ./hanoiVision -c SimonTopRight

- Kinect (on Windows 7 machine)
- Log on as chienming (Passwd: LabPassword)
- In Visual Studio, open up KinectSender.sln.
- Make sure the Kinect is connected.
- Hit the "play" button to run it.

Internal

- Speakers
    1. Connect the speakers to the new Mac Pro.
    2. Turn the speakers on by rotating the silver knob until the light is visible.
    3. On the Mac Pro, go to System Preferences > Sound > Output. Set the output device to "Headphones."
    4. Set the volume to about 90% of the bar.

C6 Controller

1. From the new Mac Pro, launch crystal.backoff.BackoffController
2. In "Main Motors Enabler" window:
    1. Click the "Send data" checkbox.
    2. Click the "no_left" presets button. (This is to avoid the ma3_j3 problems.)
    3. Click the "Update" button.

Experiment

1. Set up the cups on peg A.
2. Have subject sign consent form.
3. Walk through protocol with subject.
4. Lift the e-stop.
5. In the "Hanoi Experiment" window, select the condition (toggle the checkbox).
6. In the "Hanoi Experiment" window, hit the button "Start Game."

7. Have subject do survey on iMac.

Subject instructions

1. You're going to work together with Simon to solve the Towers of Hanoi problem.
2. The goal is to move the entire stack of 5 cups from peg A to peg C by moving one cup at a time.
3. In the entire game, you need to preserve the ordering of the cups. The ordering is: pink → orange → yellow → green → blue. For example, pink can sit on top of any other cup. But orange can't sit on top of pink.
4. You can only use one arm at a time (no holding two cups).
5. Stand in this area across from Simon.
6. An action is either picking up or placing a cup. Try not to do actions TOO quickly — that is, space them out by about two seconds.
7. That said, you should try to do the task as quickly as possible given these constraints.
8. During the interaction, Simon may ask you to do some actions – either picking up a cup, or placing it on a certain peg. (You don't have to listen, especially if the question doesn't make sense due to vision problems.)
9. If Simon has trouble manipulating an object (e.g. clearly intends to grasp or release a cup but fails, or a second cup sticks to the one he grabs), it's fine to fix it. In fact, you should, to keep the experiment going.
10. I'm going to go click a button to start the experiment, and Simon will start moving. As soon as his hand is over the table you can start doing things.

***1. Name**

***2. Gender**

- Female
- Male

***3. Do we have permission to use images or video of you in academic publications and/or conferences?**

- Yes
- No

***4. Do we have permission to use images or video of you in future studies (i.e. have a subject observe your interaction with Simon)?**

- Yes
- No

***5. On a scale from 1-100, how much did you contribute towards mentally solving the puzzle? (50 means both contributed equally)**

Your contribution

***6. On a scale from 1-100, how much did you contribute towards physically solving the puzzle? (50 means both contributed equally)**

Your contribution

***7. Please rate the following statements about the task.**

	Strongly disagree	Disagree	Slightly disagree	Neutral	Slightly agree	Agree	Strongly agree
The task would be difficult for me to complete alone.							
The task would be difficult for Simon to complete alone.							
The task was difficult for us to complete							

The task was difficult for us to complete together.							
My performance was important for completing the task.							
Simon's performance was important for completing the task.							

Comments (optional)

***8. Please rate the following statements about the interaction with Simon.**

	Strongly disagree	Disagree	Slightly disagree	Neutral	Slightly agree	Agree	Strongly agree
Simon was responsive to my actions.							
Simon was team-oriented.							
I trusted Simon's decisions.							
I had influence on Simon's behavior.							
Simon had influence on my behavior.							
I had to spend time waiting for Simon.							
Simon had to spend time waiting for me.							
We were efficient in completing the task.							
The interaction pace felt natural.							
There were awkward moments in the interaction.							

Comments (optional)

***9. Which best describes your relative roles in completing the task?**

- Simon was my superior.
- Simon was my equal.
- Simon was my subordinate.

***10. Please provide a critical review of Simon as a team member. (Imagine that Simon is being evaluated at a workplace.)**

11. Any other comments about Simon's collaboration skills or the experiment in general (optional):

Done

CONSENT DOCUMENT FOR ENROLLING PARTICIPANTS IN A RESEARCH STUDY

Georgia Institute of Technology
Project Title: Human-Robot Contingency and Turn-Taking
Protocol and Consent Title: 08/27/09v1
Investigators: Dr. Andrea Thomaz, Crystal Chao, Jeff Kiser, and Jinhan Lee

You are being asked to be a volunteer in a research study.

Purpose:
The purpose of this study is to learn more about robot technology designed to communicate with people using socially interaction.

Procedures:
If you decide to participate in this study, your part will involve:

- The study will take between 30 minutes and one hour.
- You will first be given an introduction by the experimenter. During this introduction the experimenter will explain the robot technology that will be used in the study.
- You will then be asked to socially interact with the robot technology. For example, our robot may try to get your attention or talk to you.
- We will collect audio and video data from your session. These tapes will remain confidential, and will not be distributed in any way or used for any purpose other than our analysis. The tapes will be erased after the study is finished.
- Finally, after the experiment is over, you will have the opportunity to ask questions and learn more about the goals of this research if you want to.

Risks or Discomforts:
The risks involved are no greater than those involved in daily activities such as holding conversations with people.

Benefits:
We hope that someday robots will be able to initiate and hold conversations with people. In the meantime, you may benefit from being in this study if you enjoy interacting with new technology and learning about the state of the art robotics research going on at Georgia Tech.

Compensation to You:
There is no compensation for participation.

Confidentiality:
The data collected about you will be kept private to the extent allowed by law. To protect your privacy, your records will be kept under a code number rather than by name. Your records will be kept in locked files and only study staff will be allowed to look at them. Your name and any other fact that might point to you will not appear when results of this study are presented or published.

The video and audio tapes collected as part of the evaluation will be stored in a locked cabinet in Dr. Thomaz's office. Dr. Thomaz and the students involved with the project being evaluated will have access to these tapes for the purpose of analyzing the human-robot interaction. Once the research project is completed, the tapes will be erased.

To make sure that this research is being carried out in the proper way, the Georgia Institute of Technology IRB may review study records. The Office of Human Research Protections may also look over study records during required reviews.

Costs to You:
There are no costs to you, other than your time, for being in this study.

In Case of Injury/Harm:
If you are injured as a result of being in this study, please contact Principal Investigator, Dr. Andrea Thomaz, at telephone
(404) 385-3365. Neither the Principal Investigator nor Georgia Institute of Technology has made provision for payment of costs associated with any injury resulting from participation in this study.

Participant Rights:
- Your participation in this study is voluntary. You do not have to be in this study if you don't want to be.
- You have the right to change your mind and leave the study at any time without giving any reason and without penalty.
- Any new information that may make you change your mind about being in this study will be given to you.
- You will be given a copy of this consent form to keep.
- You do not waive any of your legal rights by signing this consent form.

Questions about the Study:
- If you have any questions about the study, you may contact Dr. Andrea Thomaz at telephone (404) 385-3365.
- If you have any questions about your rights as a research subject, you may contact:

Ms. Melanie Clark, Georgia Institute of Technology
Office of Research Compliance, at (404) 894-6942.

Your Consent:
If you sign below, it means that you have read (or have had read to you) the information given in this consent form, and you would like to be a volunteer in this study.

_____
Participant Name (printed)

_____ _____

Participant Signature Date

_____ _____

Signature of Person Obtaining Consent Date

APPENDIX C

MATERIALS FOR CONTEXT-FREE OBJECT PLAY STUDY

- Participant recruitment text
- Experimenter protocol
- Post-study questionnaire
- IRB consent form

Hello!

We are inviting humans to come interact with Simon the Robot for a study this Thursday-Friday, 8/23-8/24. The study involves playing with toys and takes about 10 mins. Get your picture taken with Simon and help him improve his social skills!

The study will take place at the RIM Center in CCB. To participate, send an email to simon@mail.gatech.edu with your availability on those days.

Simon looks forward to meeting you!

Sincerely,
Crystal Chao and Andrea L. Thomaz
Socially Intelligent Machines Lab

JHRI 2013 protocol

Meka server

1. Go to the RTPC and open up a terminal, and run: m3rt_server_run
2. In a another terminal window, cd ~/mekabot/m3sim/trunk/python/scripts and run ./m3sim_demo_c6_animation_new.py

External modules

- Asus (on dokdo/GM beast)
    1. cd to ~/projects/misc/pcl/bin
    2. Run ./pcl_openni_segmentation

- Kinect (on Windows 7 machine)
    1. Log on as chienming with lab password.
    2. In Visual Studio, open up KinectSender.sln.
    3. Make sure the Kinect is connected.
    4. Hit the “play” button to run it.

- Audio (on old Mac Pro)
    1. Run ~/projects/misc/puredata/prosody.pd
    2. In a terminal window, launch crystal.perception.PureDataBroadcaster
    3. In the PureData console, click the “compute audio” checkbox.
    4. In the prosody.pd app, click the “connect” bang.

Internal

- Speakers
    1. Connect the speakers to the new Mac Pro.
    2. Turn the speakers on by rotating the silver knob until the light is visible.
    3. On the Mac Pro, go to System Preferences > Sound > Output. Set the output device to “Headphones.”
    4. Set the volume to about 90% of the bar.

C6 Controller

1. From the new Mac Pro, launch crystal.roles.RolesController
2. In “Main Motors Enabler” window:
    1. Click the “Send data” checkbox.
    2. Click the “Update” button.

Experiment

1. Set up the toys on the side posing table.
2. Have subject sign consent form.
3. Walk through protocol with subject.
4. Give microphone to subject.
5. Tell subject to turn the microphone on and test speech volume.
6. Lift the e-stop.
7. Start video recording.
8. In the “ObjectInteraction” window:
    1. Type the subject’s name/id and hit enter.
    2. Click the “Start Petri net” button. Wait for robot to assume idle pose.
    3. Click the “Start logging” button.
    4. Click the “allow engagement” button.
9. Stop the interaction after 10 mins.
    1. Click the “Stop logging” button.
    2. Stop video recording.
10. Have subject do survey on iMac.

Subject instructions

1. Stand in this spot behind the table. Don’t move the table!!
2. You’re going to play with Simon for two interaction sessions, with these two sets of objects.
3. Simon is curious about objects and wants to interact with them, and you’re supposed to teach him about them. But you’re not going to understand what he’s saying because he speaks a different language.
    1. Tell Simon about the objects. You can talk about them, tell stories, whatever you want.
    2. Simon can see your hands, head, and objects on the table in front of him. He can hear your speech through the microphone.
    3. You can treat Simon like a 4-year-old child who speaks a foreign language.
4. Try to keep engaging the robot even if you’re uncertain what’s happening.
5. When I say “ok”, wave to the robot and say “Hello Simon” to start.

***1. Name**

***2. Age**

***3. Gender**

- Female
- Male

***4. Do we have permission to use images or video of you in academic publications and/or conferences?**

- Yes
- No

***5. Do we have permission to use images or video of you in future studies (i.e. have another participant observe your interaction with Simon)?**

- Yes
- No

***6. Do you interact often with young children?**

- Yes
- No

Please describe (e.g. you are a parent, babysit, or have younger siblings):

***7. How did you find the pacing of the interaction?**

- Slow

- Medium
- Fast

✱8. Who led the interaction?

- Simon
- Me
- About equal

✱9. Please rate the following statements about the interaction with Simon.

	Strongly disagree	Disagree	Slightly disagree	Neutral	Slightly agree	Agree	Strongly agree
Simon was responsive to my actions.							
I had influence on Simon's behavior.							
Simon had influence on my behavior.							
Simon listened to me.							
Simon talked over or interrupted me.							
I had to spend time waiting for Simon.							
Simon had to spend time waiting for me.							
The interaction pace felt natural.							
There were silences where nothing happened.							
There were overlaps where we both tried to act.							
There were awkward moments in the interaction.							

Comments (optional)

✱10. How would you classify...

	Strongly introverted	Introverted	Slightly introverted	Neutral	Slightly extroverted	Extroverted	Strongly extroverted
... Simon's personality?							

... your own personality?

***11. List some adjectives describing Simon's personality:**

***12. Please provide a critical review of Simon's social skills.**

13. Other comments (optional):

Done

CONSENT DOCUMENT FOR ENROLLING PARTICIPANTS IN A RESEARCH STUDY

Georgia Institute of Technology
Project Title: Adapative Turn-taking Routines for HRI via Stereotype Models
Consent Title: 03/27/2012 v1
Investigators: Dr. Andrea Thomaz, Dr. Alan Wagner

You are being asked to be a volunteer in a research study.

Sponsor: This study is sponsored by the Office of Naval Research.

Purpose:
We are interested in enabling robots to successfully engage human partners in reciprocal turn-taking interactions.

Procedures:
If you decide to participate in this study, your part will involve:

- The study will take between 30 minutes and one hour.
- You will first be given an introduction by the experimenter. During this introduction the experimenter will explain the robot technology that will be used in the study.
- You will then be asked to socially interact with the robot technology. For example, you may play a simple game like Simon Says or build something together with blocks.
- We will collect audio and video data from your session. These tapes will remain confidential, and will not be distributed in any way or used for any purpose other than our analysis. The tapes will be erased after the research is finished.
- Finally, after the experiment is over, you will have the opportunity to ask questions and learn more about the goals of this research if you like.

Risks or Discomforts:
The risks involved are no greater than those involved in daily activities such as holding conversations with people.

Benefits:
We hope that someday robots will be able to initiate and hold conversations with people. In the meantime, you may benefit from being in this study if you enjoy interacting with new technology and learning about the state of the art robotics research going on at Georgia Tech.

Compensation to You:
You will receive $10/hour for your time, either cash or a gift card.

Confidentiality:
The data collected about you will be kept private to the extent allowed by law. To protect your privacy, your records will be kept under a code number rather than by name. Your records will be kept in locked files and only research staff will be allowed to look at them. Your name and any other fact that might point to you will not appear when results of this study are presented or published.

The video and audio data collected as part of the evaluation will be kept on a storage device in a locked cabinet in Dr. Thomaz's office. Dr. Thomaz and the students involved with the project being evaluated will have access to these tapes for the purpose of analyzing the human-robot interaction. Once the research project is completed, the data will be erased. The only exception to this is if you agree to the video release at the end of this form, then we may use video clips of your interaction with the robot in our conference presentations or in research videos about the project that will be posted on our website.

To make sure that this research is being carried out in the proper way, the Georgia Institute of Technology IRB may review study records. The Office of Human Research Protections and the Office of Naval Research may inspect the investigator's records in order to ensure compliance with federal laws.

Costs to You:
There are no costs to you, other than your time, for being in this study.

In Case of Injury/Harm:
If you are injured as a result of being in this study, please contact Principal Investigator, Dr. Andrea Thomaz, at telephone (404) 385-3365. Neither the Principal Investigator nor Georgia Institute of Technology has made provision for payment of costs associated with any injury resulting from participation in this study.

Participant Rights:
- Your participation in this study is voluntary. You do not have to be in this study if you don't want to be.
- You have the right to change your mind and leave the study at any time without giving any reason and without penalty.
- Any new information that may make you change your mind about being in this study will be given to you.

- You will be given a copy of this consent form to keep.
- You do not waive any of your legal rights by signing this consent form.

Questions about the Study:

- If you have any questions about the study, you may contact Dr. Andrea Thomaz at telephone (404) 385-3365.
- If you have any questions about your rights as a research subject, you may contact:

Ms. Melanie Clark, Georgia Institute of Technology
Office of Research Compliance, at (404) 894-6942.

Your Consent:

If you sign below, it means that you have read (or have had read to you) the information given in this consent form, and you would like to be a volunteer in this study.

_____
Participant Name (printed)

_____ _____
Participant Signature Date

_____ _____
Signature of Person Obtaining Consent Date

Video Release

I consent to the use of video recordings from this study in conference presentations and research promotion videos to be displayed on a project webpage.

Initial_____

APPENDIX D

MATERIALS FOR LEGO COLLABORATION STUDY

- Participant recruitment text
- Participant instructions
- Reference sheets for participants during interaction
- Experimenter protocol
- Post-study questionnaire
- IRB consent form

Hello!

We are inviting volunteers to interact with Simon the social robot for a study this Wednesday-Friday, 1/7-1/9. The study involves collaborating with Simon to build a block model, and we estimate it should take about 30 mins of your time.

To participate, send me a mail (cchao@gatech.edu) with your availability on those days. The study will take place at the RIM Center in CCB.

Simon looks forward to working with you!

Sincerely,
Crystal Chao and Andrea L. Thomaz
Socially Intelligent Machines Lab

1 Overview

In this study you and Simon will design and build a house together out of blocks. You will each have different but compatible specifications for the shared design, so you will need to talk to Simon and work together to decide what to build. However, Simon will not always understand your speech or your actions. You have a maximum of 15 minutes to complete the interaction. If you build the house correctly within that time, you will be eligible to win a $50 Amazon gift certificate.

Assemble the model by placing blocks onto the green building plate. When you are done or wish to stop, press the large button on the right side of the table to submit your entry. You will then be asked to fill out a survey about your experience interacting with the robot.

2 Model specifications

You can assume that Simon already knows all the rules in this section. The house will be made of a single flat layer of blocks on the building plate, viewed from above from your perspective. It needs to have 1 **roof**, 1 **wall**, 1 **door**, and 1 **window** (Figure 1(a)). Each part can only be a single color, and any directly adjacent parts may not be the same color. The model has to be fully filled with blocks; there cannot be any holes.

2.1 Wall

The "wall" is a rectangle beneath the roof. The color of the house is the color of the wall. The wall in Figure 1(a) is yellow, 5 units wide, and 3 units tall.

2.2 Roof

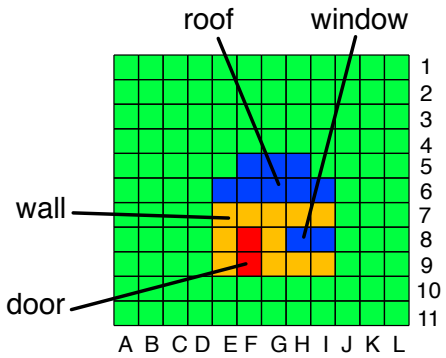
The roof is over the wall and must be the same width as the wall. A roof is widest at the bottom and narrowest at the top. From bottom to top, the roof decreases in width by exactly one unit on each side (see Figure 1(b)). The roof in Figure 1(a) is blue, 5 units wide, and 2 units tall.

2.3 Door

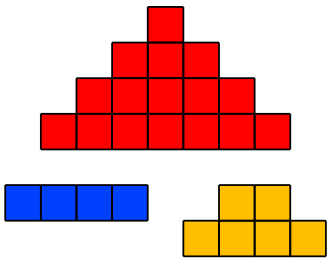
A door is a rectangle inside of a wall. It must touch the bottom of the wall and cannot touch the roof or the window. The door in Figure 1(a) is red, 1 unit wide, and 2 units tall.

2.4 Window

A window is also a rectangle inside of a wall, but cannot touch the bottom or the top of a wall. Doors and windows can't touch. The window in Figure 1(a) is blue, 2 units wide, and 1 unit tall.



(a) A yellow house that is 5 units wide by 5 units tall.



(b) Roofs decrease in width by one unit on each side from bottom to top.

Figure 1: Examples of valid components: (a) a house, and (b) different roofs.

3 Your design specifications

In addition to being a valid house according to the previous section, your model design needs to satisfy all of the following requirements and all of Simon's requirements. Since you and Simon start off not knowing each other's specifications, you will need to work together to agree on a design.

1. The window has to be blue.
2. The roof height has to be greater than the door height.
3. The window has to be left of the door.
4. The house has to be to the right of column C.
5. The door and the window have to be the same width.

Make sure you understand your own requirements and ask the experimenter clarifying questions before starting the interaction, as after it starts you may not speak to anyone other than Simon until it is finished.

4 Building guidelines

1. Keep the outermost edge of the green building plate empty (i.e. columns A and L, rows 1 and 11).
2. Simon won't push blocks into the plate, so you will have to help. Simon also cannot remove blocks from the plate, but you can.
3. Simon can't see or use the blocks on the far ends of the table, but you can move blocks there to the center area of the table for Simon to use. Simon prefers to use larger blocks and cannot use 1x1 blocks.
4. Blocks out in the center area of the table should be kept several inches apart so Simon can grab them.
5. Simon doesn't place blocks accurately, so take your best guess as to what he meant to do.
6. Stay behind the red line on the ground when you're not using blocks.

5 What Simon can understand

Simon only understands a limited set of phrases. After you have read this document through, you will practice speaking them with the experimenter's guidance. When you have the headset on, read the following lines into the headset microphone until the text on the screen matches what you said:

- "Hello Simon."
- "What color is the window?"
- "How tall should the house be?"
- "Should the roof and the door be the same height?"
- "The door color and the window color must be different."
- "Can the wall be three units high?"
- "Let's make it red."
- "Yes." / "Okay." / "Sure." / "No." / "I don't know."
- "That is the door."
- "What?" / "Can you say that again?"
- State each of the design specifications that you are given.
- Ask each of your design specifications as a question.
- Ask about, then suggest a value for: the house width, the roof color, the wall height

  You may then try speaking similar phrases to see if Simon will understand them.

6 Starting the interaction

Start by standing behind the red line, facing Simon. The experimenter will turn the robot on and move its arms into place. When Simon's ear lights turn on, say "hello" to Simon. The 15-minute timer does not start until Simon says "hello" back to you. This may take several tries if Simon does not hear or understand you. Simon will not understand anything else you say until he says "hello" back.

When the interaction starts, the experimenter will start a 15-minute timer on an iPad and place it at the end of the table on your left side. You may refer to it to monitor your progress.

Once the interaction has started, do not communicate with the experimenter or any other people until it has completed. End the interaction by hitting the button on the right side of the table.

Start	Hello Simon.
Statements	The window has to be blue. Let's make the window blue.
Questions	Can the window be blue? What color should the window be? What? / Can you repeat that?
Answers	Yes. / No. / I don't know. That is the window.

The window has to be blue.
The roof height has to be greater than the door height.
The window has to be left of the door.
The house has to be to the right of column C.
The door and the window have to be the same width.

Verbal instructions to users

- When not doing anything on the table, try to stand behind the taped line on the ground.
- Simon can't see or reach the blocks on the sides... you will have to move them to the middle region if you want him to use them.
- Keep the objects apart from each other on the table.
- Place them in the long orientation along the table, easier for Simon to pick up.
- Don't give Simon 1x1s blocks. (Bigger is better)
- Simon won't push the blocks into the mat, so you will have to.
- Simon cannot remove blocks from the mat.
- There's error with Simon's placement, so take your best guess as to what he meant to do.
- Start by saying "hello," when the robot also responds with "hello" the timer starts.
- When pointing at the mat, don't touch the mat.

Start of the day

- check time server
- check robot works
- check audio out to speakers

User explanation

- sign IRB consent form
- give instructions to user, tell to read through in front of the table
- answer questions about task
- grammar familiarization with headset
- answer questions about grammar

Start robot interaction

- run Meka server, c6 script
- run the KinectSender
- on dokdo, run ./segment_blocks for perception
- make sure TTS is running (run from MacSpeechSynthesis Xcode project)
- launch domains.blocks.demos.HouseCollabController
- connect USB for Sennheiser SD Pro 1 headset. disconnect from power, interferes with ASR
- run Inpro

- check speaker volume level
- Experiment window:
    - select the condition
    - click “log" checkbox
    - type the subject name
    - check “listening"
    - lift e-stop
    - click “both” button to ready both arms
- RECORD FROM THE CAMCORDER
    - click “start" button
- start iPad timer, place on side table

End of interaction

- click “stop logging” button
- STOP THE CAMCORDER
- get headset back from user
- set up user with survey
- put robot in safe pose to e-stop
- charge headset from power adapter (must unplug USB connection to work)

End of the day

- import video and back up
- shut down KinectSender
- shut down Meka server
- shut down perception
- to generate overhead movies, run on each subject:
    ./export_times [subject_name]
    ffmpeg -f concat -i durations.txt -vb 20M out.mp4
    - back up to drive through Mac and sshfs
- back up data

Lego Collaboration Study

∗1. Name

∗2. Age

∗3. Gender

- Female
- Male

∗4. Do we have permission to use images or video of you in academic publications and/or conferences?

- Yes
- No

∗5. Do we have permission to use images or video of you in future studies (i.e. have another participant observe your interaction with Simon)?

- Yes
- No

∗6. How did you find the pacing of the interaction?

Very slow	Slow	Slightly slow	Medium	Slightly fast	Fast	Very fast

∗7. Who led the interaction?

- Simon
- Me
- About equal

∗8. On a scale from 1-100, how much did you contribute towards the task solution? (50 means both contributed equally)

∗9. Did you complete the task successfully?

- yes
- no
- I don't know

Explanation (optional)

∗10. Please rate the following statements about the interaction with Simon.

	Strongly disagree	Disagree	Slightly disagree	Neutral	Slightly agree	Agree	Strongly agree
Simon and I were on the same page.							
Simon was team-oriented.							
Our team worked fluently together.							
Our team's fluency improved over time.							
Simon was responsive to my actions.							
Simon listened to me.							
Simon talked over or interrupted me.							
I could tell whether or not Simon heard me.							
I could tell whether or not Simon understood me.							
I had to spend time waiting for Simon.							
Simon had to spend time waiting for me.							
There were awkward moments in the interaction.							

Comments (optional)

∗11. Please provide a critical review of Simon as a team member.

12. Other comments (optional):

Done

REFERENCES

[1] Allen, J. F., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W., "PLOW: A collaborative task learning agent," in *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI)*, pp. 1514–1519, 2007.

[2] Allen, J. F., Ferguson, G., and Stent, A., "An architecture for more realistic conversational systems," in *Proceedings of 6th International Conference on Intelligent User Interfaces (IUI)*, pp. 1–8, 2001.

[3] Arkin, R. C., *Behavior-based robotics*. MIT Press, 1998.

[4] Austin, J. L., *How to Do Things With Words*. Harvard University Press, 1962.

[5] Badalamenti, A. F., Beebe, B., Jaffe, J., Marquette, L., Helbraun, E., Andrews, H., and Ellman, L., "Poisson regulation in mother-infant gaze systems," *Mathematical and Computer Modelling*, vol. 39, pp. 305–324, 2004.

[6] Baddeley, A. D. and Hitch, G., "Working memory," in *The psychology of learning and motivation: Advances in research and theory* (Bower, G. H., ed.), pp. 47–89, Academic Press, 1974.

[7] Bangerter, A. and Clark, H. H., "Navigating joint projects with dialogue," *Cognitive Science*, vol. 27, pp. 195–225, 2003.

[8] Barrett, L. R., *An Architecture for Structured, Concurrent, Real-Time Action*. PhD thesis, University of California, Berkeley, 2010.

[9] Baumann, T. and Schlangen, D., "The InproTK 2012 release," in *In: Proceedings of the NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD)*, pp. 29–32, 2012.

[10] Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M., and Tomlinson, B., "Integrated learning for interactive synthetic characters," in *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 417–426, 2002.

[11] Bohus, D., *Error Awareness and Recovery in Conversational Spoken Language Interfaces*. PhD thesis, Carnegie Mellon University, 2007.

[12] Bohus, D. and Horvitz, E., "Facilitating multiparty dialog with gaze, gesture, and speech," in *Proceedings of the 12th International Conference on Multimodal Interfaces (ICMI)*, 2010.

[13] Bohus, D. and Horvitz, E., "Decisions about turns in multiparty conversation: From perception to action," in *Proceedings of the 13th International Conference on Multimodal Interfaces (ICMI)*, pp. 153–160, 2011.

[14] Bohus, D. and Rudnicky, A., "RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda," in *INTERSPEECH*, 2003.

[15] Bratman, M., "Shared cooperative activity," *The Philosophical Review*, vol. 101, no. 2, pp. 327–341, 1992.

[16] Breazeal, C., "Emotion and sociable humanoid robots," *International Journal of Human-Computer Studies*, vol. 59, no. 1–2, pp. 119–155, 2003.

[17] Brooks, R., "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[18] Broz, F., *Planning for Human-Robot Interaction: Representing Time and Human Intention*. PhD thesis, Carnegie Mellon University, 2008.

[19] Burgoon, J., Stern, L., and Dillman, L., *Interpersonal Adaptation: Dyadic Interaction Patterns*. Cambridge University Press, 1995.

[20] Cakmak, M., Chao, C., and Thomaz, A. L., "Designing interactions for robot active learners," *IEEE Transactions on Autonomous Mental Development*, vol. 2, pp. 108–118, June 2010.

[21] Cao, T. and Anderson, A. C., "A fuzzy Petri net approach to reasoning about uncertainty in robotic systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 317–322, 1993.

[22] Cassell, J., Bickmore, T., Campbell, L., Chang, K., Vilhjálmsson, H., and Yan, H., "Requirements for an architecture for embodied conversational characters," in *Computer Animation and Simulation*, pp. 109–120, Springer Verlag, 1999.

[23] Cassell, J. and Thorissón, K., "The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents," *Applied Artificial Intelligence*, vol. 13, pp. 519–538, 1999.

[24] Chao, C., Cakmak, M., and Thomaz, A., "Transparent active learning for robots," in *Proceedings of the 5th ACM/IEEE international Conference on Human-Robot Interaction (HRI)*, pp. 317–324, 2010.

[25] Chao, C., Lee, J. H., Begum, M., and Thomaz, A., "Simon plays Simon says: The timing of turn-taking in an imitation game," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 235–240, 2011.

[26] Chao, C. and Thomaz, A., "Timing in multimodal reciprocal interactions: Control and analysis using timed Petri nets," *Journal of Human-Robot Interaction*, vol. 1, no. 1, pp. 4–25, 2012.

[27] Chao, C. and Thomaz, A., "Controlling social dynamics with a parametrized model of floor regulation," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 4–29, 2013.

[28] Clancy, P. M., Thompson, S. A., Suzuki, R., and Tao, H., "The conversational use of reactive tokens in English, Japanese and Mandarin," *Journal of Pragmatics*, vol. 26, pp. 355–387, 1996.

[29] Clark, H. H. and Brennan, S. E., "Grounding in communication," in *Perspectives on Socially Shared Cognition* (Resnick, L. B., Levine, J. M., and Teasley, S. D., eds.), pp. 127–149, American Psychological Association, 1991.

[30] Clark, H. H. and Krych, M. A., "Speaking while monitoring addressees for understanding," *Journal of Memory and Language*, vol. 50, no. 1, pp. 62–81, 2004.

[31] Clark, H. H. and Tree, J. E. F., "Using *uh* and *um* in spontaneous speaking," *Cognition*, vol. 84, pp. 73–111, 2002.

[32] Dantam, N. and Stilman, M., "The motion grammar: Analysis of a linguistic method for robot control," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 704–718, 2013.

[33] Deits, R., Tellex, S., Thaker, P., Simeonov, D., Kollar, T., and Roy, N., "Clarifying commands with information-theoretic human-robot dialog," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 58–79, 2013.

[34] DeVault, D., Sagae, K., and Traum, D., "Can I finish? Learning when to respond to incremental interpretation results in interactive dialogue," in *Proceedings of the 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGdial)*, pp. 11–20, 2009.

[35] Duncan, S., "On the structure of speaker-auditor interaction during speaking turns," *Language in Society*, vol. 3, no. 2, pp. 161–180, 1974.

[36] Feil-Seifer, D. and Mataric, M., "A Simon-says robot providing autonomous imitation feedback using graded cueing," in *Proceedings of the International Meeting for Autism Research (IMFAR)*, 2012.

[37] Ferguson, G. and Allen, J. F., "TRIPS: An integrated intelligent problem-solving assistant," in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pp. 567–572, 1998.

[38] Fillmore, C. J., "Frame semantics and the nature of language," in *Origins and Evolution of Language and Speech* (et al., S. R. H., ed.), pp. 20–32, New York Academy of Sciences, 1976.

[39] Ginzburg, J., *The Interactive Stance*. Oxford University Press, 2012.

[40] Goddeau, D., Meng, H., Polifroni, J., Seneff, S., and Busayapongchai, S., "A form-based dialogue manager for spoken language applications," in *Proceedings of the 4th International Conference on Spoken Language (ICSLP)*, vol. 2, pp. 701–704, 1996.

[41] Gombolay, M. C., Wilcox, R., and Shah, J., "Fast scheduling of multi-robot teams with temporospatial constraints," in *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[42] Grosz, B. J. and Sidner, C. L., "Attention, intentions, and the structure of discourse," *Computational Linguistics*, vol. 12, no. 3, pp. 175–204, 1986.

[43] Grosz, B. J. and Sidner, C. L., "Plans for discourse," in *Intentions in Communications* (Cohen, P. R., Morgan, J., and Pollack, M., eds.), pp. 417–444, MIT Press, 1990.

[44] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2015.

[45] Hoffman, G., *Ensemble: Fluency and Embodiment for Robots Acting with Humans*. PhD thesis, Massachusetts Institute of Technology, 2007.

[46] Hoffman, G. and Breazeal, C., "Collaboration in human-robot teams," in *Proceedings of the 1st AIAA Intelligent Systems Conference*, 2004.

[47] Hoffman, G. and Breazeal, C., "Effects of anticipatory action on human-robot teamwork: Efficiency, fluency, and perception of team," in *Proceedings of the 3rd ACM/IEEE Conference on Human-Robot Interaction (HRI)*, 2007.

[48] Holroyd, A., Rich, C., Sidner, C., and Ponsler, B., "Generating connection events for human-robot collaboration," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2011.

[49] Holroyd, A. G., "Generating engagement behaviors in human-robot interaction," Master's thesis, Worcester Polytechnic Institute, 2011.

[50] Ishii, R., Miyajima, T., Fujita, K., and Nakano, Y., "Avatars gaze control to facilitate conversational turn-taking in virtual-space multi-user voice chat system," in *Proceedings of the 6th International Conference on Intelligent Virtual Agents (IVA)*, pp. 458–458, 2006.

[51] Johnson-Roberson, M., Bohg, J., Skantze, G., Gustafson, K., Carlson, R., Rasolzadeh, B., and Kragic, D., "Enhanced visual scene understanding through human-robot dialog," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3342–3348, 2011.

[52] Johnstone, K., *Impro: Improvisation and the Theatre*. New York: Routledge, 1987.

[53] Jurafsky, D., Shriberg, E., Fox, B., and Curl, T., "Lexical, prosodic, and syntactic cues for dialog acts," in *Proceedings of the ACL/COLING-98 Workshop on Discourse Relations and Discourse Markers*, pp. 114–120, 1998.

[54] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R., "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[55] Kanda, T., Ishiguro, H., Imai, M., and Ono, T., "Development and evaluation of interactive humanoid robots," in *Proceedings of the IEEE*, vol. 92, pp. 1839–1850, 2004.

[56] Kendon, A., *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004.

[57] Kose-Bagci, H., Dautenhan, K., and Nehaniv, C. L., "Emergent dynamics of turn-taking interaction in drumming games with a humanoid robot," in *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 346–353, 2008.

[58] Kovotsky, K., Hayes, J., and Simon, H., "Why are some problems hard? Evidence from Tower of Hanoi," *Cognitive Psychology*, vol. 17, pp. 248–294, April 1985.

[59] Kozima, H., Michalowski, M. P., and Nakagawa, C., "Keepon: A playful robot for research, therapy, and entertainment," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 3–18, 2009.

[60] Krause, E., Cantrell, R., Potapova, E., Zillich, M., and Scheutz, M., "Incrementally biasing visual search using natural language input," in *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

[61] Kruijff, G. M., Zender, H., Jensfelt, P., and Christensen, H. I., "Situated dialogue and spatial organization: What, where... and why," *International Journal of Advanced Robotic Systems*, vol. 4, no. 2, pp. 125–138, 2007.

[62] Lacerda, B. and Lima, P., "Designing Petri net supervisors from LTL specifications," in *Proceedings of Robotics: Science and Systems (RSS)*, 2011.

[63] Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., Warmuth, M., and Wolf, P., "The CMU Sphinx-4 speech recognition system," in *Proceedings of 29th IEEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[64] Langacker, R. W., *Concept, Image, and Symbol: The Cognitive Basis of Grammar*. Mouton de Gruyter, 1990.

[65] Lee, J., Kiser, J. F., Bobick, A. F., and Thomaz, A. L., "Vision-based contingency detection," in *Proceedings of the 6th ACM/IEEE Conference on Human-Robot Interaction (HRI)*, 2011.

[66] Leiser, R. G., "Improving natural language and speech interfaces by the use of metalinguistic phenomena," *Applied Ergonomics*, vol. 20, no. 3, pp. 168–173, 1989.

[67] Lemaignan, S., Ros, R., Sisbot, E. A., Alami, R., and Beetz, M., "Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction," *International Journal of Social Robotics*, vol. 4, no. 2, pp. 181–199, 2012.

[68] Lison, P., "Probabilistic dialogue models with prior domain knowledge," in *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGdial)*, pp. 179–188, 2012.

[69] Matsuyama, Y., Akiba, I., Fujie, S., and Kobayashi, T., "Four-participant group conversation: A facilitation robot controlling engagement density as the fourth participant," *Computer Speech and Language*, vol. 33, no. 1, pp. 1–24, 2015.

[70] Matuszek, C., Herbst, E., Zettlemoyer, L., and Fox, D., "Learning to parse natural language commands to a robot control system," in *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2012.

[71] Mazur, A. and Cataldo, M., "Dominance and deference in conversation," *Journal of Social and Biological Systems*, vol. 12, no. 1, pp. 87–99, 1989.

[72] McGurk, H. and MacDonald, J., "Hearing lips and seeing voices," *Nature*, vol. 264, no. 5588, pp. 746–648.

[73] Meena, R., Skantz, G., and Gustafson, J., "A data-driven model for timing feedback in a map task dialogue system," in *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGdial)*, 2013.

[74] Minsky, M., *The Society of Mind*. Simon and Schuster, 1988.

[75] Mondada, L., "Multimodal resources for turn-taking: pointing and the emergence of possible next speakers," *Discourse Studies*, vol. 9, no. 2, pp. 194–225, 2007.

[76] Moon, A., Parker, C., Croft, E., and der Loos, H. V., "Design and impact of hesitation gestures during human-robot resource conflicts," *Journal of Human-Robot Interaction*, vol. 2, no. 3, pp. 18–40, 2013.

[77] Morency, L. P., de Kok, I., and Gratch, J., "A probabilistic multimodal approach for predicting listener backchannels," *Autonomous Agents and Multi-Agent Systems*, vol. 20, no. 1, pp. 70–84.

[78] Movellan, J. R., "An Infomax controller for real time detection of social contingency," in *Proceedings of the 4th International Conference on Development and Learning (ICDL)*, pp. 19–24, 2005.

[79] Murata, T., "Petri nets: Properties, analysis and applications," in *Proceedings of the IEEE*, vol. 77, pp. 541–580, 1989.

[80] Mutlu, B., *Designing Gaze Behavior for Humanlike Robots*. PhD thesis, Carnegie Mellon University, 2009.

[81] Mutlu, B., Shiwa, T., Ishiguro, T. K. H., and Hagita, N., "Footing in human-robot conversations: How robots might shape participant roles using gaze cues," in *Proceedings of the 4th ACM/IEEE Conference on Human-Robot Interaction (HRI)*, 2009.

[82] Nakano, M., Hasegawa, Y., Funakoshi, K., Takeuchi, J., Torii, T., Nakadai, K., Kanda, N., Komatani, K., Okuno, H. G., and Tsujino, H., "A multi-expert model for dialogue and behavior control of conversational robots and agents," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 248–256, 2011.

[83] Nakano, M., Hasegawa, Y., Nakadai, K., Nakamura, T., Takeuchi, J., Torii, T., Tsujino, H., Kanda, N., and Okuno, H., "A two-layer model for behavior and dialogue planning in conversational service robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3329–3335, 2005.

[84] Nakano, Y. and Fukuhara, Y., "Estimating conversational dominance in multiparty interaction," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI)*, pp. 77–84, 2012.

[85] Nooraei, B., Rich, C., and Sidner, C. L., "A real-time architecture for embodied conversational agents: Beyond turn-taking," in *Proceedings of the 7th International Conference on Advances in Computer-Human Interactions (ACHI)*, 2014.

[86] Oreström, B., *Turn-taking in English conversation*. CWK Gleerup, 1983.

[87] Puckette, M., "Pure Data: another integrated computer music environment," in *Proceedings of the International Computer Music Conference*, pp. 37–41, 1996.

[88] Raux, A., *Flexible Turn-Taking for Spoken Dialog Systems*. PhD thesis, Carnegie Mellon University, 2008.

[89] Raux, A. and Eskenazi, M., "A finite-state turn-taking model for spoken dialog systems," in *Proceedings of Human Language Technologies (HLT)*, 2009.

[90] Raux, A. and Eskenazi, M., "Optimizing the turn-taking behavior of task-oriented spoken dialog systems," *ACM Transactions on Speech and Language Processing*, vol. 9, no. 1, pp. 1–23, 2012.

[91] Rich, C., Ponsler, B., Holroyd, A., and Sidner, C. L., "Recognizing engagement in human-robot interaction," in *Proceedings of the 5th ACM/IEEE Conference on Human-Robot Interaction (HRI)*, 2010.

[92] Rose, R. C. and Kim, H. K., "A hybrid barge-in procedure for more reliable turn-taking in human-machine dialog systems," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 198–203, 2003.

[93] Rosenthal, S. and Veloso, M., "Modeling humans as observation providers using POMDPs," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 53–58, 2011.

[94] Sacks, H., Schegloff, E., and Jefferson, G., "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, pp. 696–735, 1974.

[95] Sakita, K., Ogawara, K., Murakami, S., Kawamura, K., and Ikeuchi, K., "Flexible cooperation between human and robot by interpreting human intention from gaze information," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 846–851, 2004.

[96] Salem, M., Kopp, S., Wachsmuth, I., Rohlfing, K., and Joublin, F., "Generation and evaluation of communicative robot gesture," *International Journal of Social Robotics*, vol. 4, pp. 201–207, 2012.

[97] Schegloff, E. A., "Recycled turn beginnings: A precise repair mechanism in conversation's turn-taking organization," in *Talk and Social Organisation* (Button, G. and Lee, J. R. E., eds.), pp. 70–85, Clevedon, England: Multilingual Matters, 1987.

[98] Schegloff, E. A., "Overlapping talk and the organization of turn-taking for conversation," *Language in Society*, vol. 29, no. 1, pp. 1–63, 2000.

[99] Schegloff, E. A., Jefferson, G., and Sacks, H., "The preference for self-correction in the organization of repair in conversation," *Language*, vol. 53, no. 2, pp. 361–382, 1977.

[100] Schegloff, E. A. and Sacks, H., "Opening up closings," *Semiotica*, vol. 8, pp. 289–327, 1973.

[101] Schlangen, D. and Skantze, G., "A general, abstract model of incremental dialogue processing," *Dialogue and Discourse*, vol. 2, no. 1, pp. 83–111, 2011.

[102] Searle, J. R., *Speech Acts: An Essay in the Philosophy of Language*, vol. 20. Cambridge University Press, 1969.

[103] Selfridge, E. and Heeman, P., "Importance-Driven Turn-Bidding for spoken dialogue systems," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 177–185, 2010.

[104] Shah, J., Wiken, J., Williams, B., and Breazeal, C., "Improved human-robot team performance using Chaski, a human-inspired plan execution system," in *Proceedings of the 6th International Conference on Human-Robot Interaction (HRI)*, pp. 29–36, 2011.

[105] She, L., Yang, S., Cheng, Y., Jia, Y., Chai, J. Y., and Xi, N., "Back to the blocks world: Learning new actions through situated human-robot dialogue," in *Proceedings of 15th Meeting on Discourse and Dialogue (SIGdial)*, 2014.

[106] Sidner, C. L., "An artificial discourse language for collaborative negotiation," in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, pp. 814–819, 1994.

[107] Steedman, M. and Baldridge, J., "Combinatory categorial grammar," in *NonTransformational Syntax: Formal and Explicit Models of Grammar*, pp. 181–224, Wiley-Blackwell, 2011.

[108] Ström, N. and Seneff, S., "Intelligent barge-in in conversational systems," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.

[109] ten Bosch, L., Oostdijk, N., and de Ruiter, J. P., "Durational aspects of turn-taking in spontaneous face-to-face and telephone dialogues," in *Text, Speech and Dialogue* (Sojka, P., Kopecek, I., and Pala, K., eds.), vol. 3206 of *Lecture Notes in Computer Science*, pp. 563–570, 2004.

[110] Thomaz, A. and Chao, C., "Turn-taking based on information flow for fluent human-robot interaction," *AI Magazine Special Issue on Dialogue With Robots*, vol. 32, no. 4, pp. 53–63, 2011.

[111] Thorissón, K., Gislason, O., Jonsdottir, G., and Thorissón, H., "A multiparty multimodal architecture for realtime turntaking," in *Intelligent Virtual Agents* (Allbeck, J., Badler, N., Bickmore, T., Pelachaud, C., and Safonova, A., eds.), vol. 6356 of *Lecture Notes in Computer Science*, pp. 350–356, Springer Berlin Heidelberg, 2010.

[112] Tomasello, M., *Origins of Human Communication*. MIT Press, 2008.

[113] Traum, D., DeVault, D., Lee, J., Wang, Z., and Marsella, S., "Incremental dialogue understanding and feedback for multiparty, multimodal conversation," in *Intelligent Virtual Agents (IVA)*, vol. 7502, pp. 275–288, 2012.

[114] Trevarthen, C., "Communication and cooperation in early infancy: A description of primary intersubjectivity," in *Before Speech: The Beginning of Interpersonal Communication* (Bullowa, M., ed.), pp. 389–450, Cambridge University Press, 1979.

[115] Trevor, A. J. B., "Fast segmentation of organized point cloud data," in *Proceedings of the International Conference on Robotics and Automation (ICRA), Advanced 3D Point Cloud Processing with Point Cloud Library (PCL)*, 2012.

[116] Tronick, E., Als, H., and Adamson, L., "Structure of early face-to-face communicative interactions," in *Before Speech: The Beginning of Interpersonal Communication* (Bullowa, M., ed.), pp. 349–374, Cambridge: Cambridge University Press, 1979.

[117] van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., and Barros, A., "Workflow patterns," *Distributed and Parallel Databases*, vol. 14, no. 3, pp. 5–51, 2003.

[118] Wang, J., *Timed Petri Nets: Theory and Application*. Springer, 1998.

[119] Wang, Z. and Lemon, O., "Time-dependent infinite POMDPs for planning real-world multimodal interactions," in *ESSLLI Workshop on Formal and Computational Approaches to Multimodal Communication*, 2012.

[120] Warneken, F., Lohse, K., Melis, A., and Tomasello, M., "Young children share the spoils after collaboration," *Psychological Science*, vol. 22, pp. 267–73, 2011.

[121] Weinberg, G. and Blosser, B., "A leader-follower turn-taking model incorporating beat detection in musical human-robot interaction," in *Proceedings of the 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 227–228, 2009.

[122] Weinberg, G. and Driscoll, S., "Robot-human interaction with an anthropomorphic percussionist," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI)*, pp. 1229–1232, 2006.

[123] Whittaker, S. and Stenton, P., "Cues and control in expert-client dialogues," in *26th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 123–130, 1988.

[124] Wiemann, J. M., "Explication and test of a model of communicative competence," *Human Communication Research*, vol. 3, no. 3, pp. 195–213, 1977.

[125] Williams, J. and Young, S., "Partially observable Markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 231–422, 2007.

[126] Wimmer, H. and Perner, J., "Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception," *Cognition*, vol. 13, no. 1, pp. 103–128, 1983.

[127] Wittgenstein, L., *Philosophical Investigations*. Blackwell Publishing, 1953.

[128] Yngve, V., "On getting a word in edgewise," in *Papers from the Sixth Regional Meeting of the Chicago Linguistic Society*, pp. 567–577, 1970.

[129] Yu, C., Smith, T. G., Hidaka, S., Scheutz, M., and Smith, L. B., "A data-driven paradigm to understand multimodal communication in human-human and human-robot interaction," in *Proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA)*, pp. 232–244, 2010.

[130] Zhang, W., "Representation of assembly and automatic robot planning by Petri net," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 2, pp. 418–422, 1989.